

UNIVERSIDADE FEDERAL DE SANTA CATARINA – UFSC
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO

FRANCISCO DAS CHAGAS MENDES DOS SANTOS

Um modelo de PARSER para aplicação em ambientes de
Projeto de Sistemas Mecânicos

Dissertação submetida à Universidade Federal de Santa Catarina como requisito final para a obtenção do título de Mestre em Ciência da Computação.

Profº Ilson Wilmar Rodrigues Filho, Dr.
Orientador

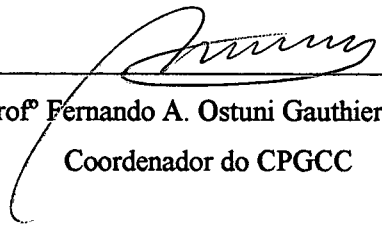
Profº João Bosco da Mota Alves, Dr.
Co-Orientador

Florianópolis, Agosto de 2002

Um modelo de PARSER para aplicação em ambientes de Projeto de Sistemas Mecânicos

FRANCISCO DAS CHAGAS MENDES DOS SANTOS

Esta dissertação foi julgada adequada para a obtenção do título de MESTRE EM CIÊNCIA DA COMPUTAÇÃO, na área de concentração SISTEMAS DE CONHECIMENTO, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

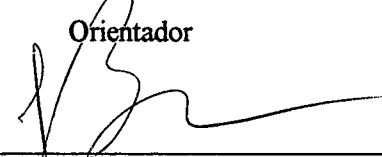


Prof^o Fernando A. Ostuni Gauthier, Dr.
Coordenador do CPGCC

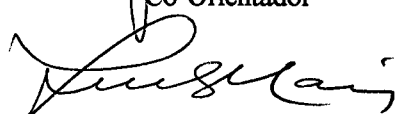
BANCA EXAMINADORA:



Prof^o Ilson Wilmar Rodrigues Filho, Dr.
Orientador



Prof^o João Bosco da Mota Alves, Dr.
Co-Orientador



Prof^o Luiz Fernando Jacintho Maia, Dr.
Examinador

DEDICATÓRIA

Em memória

da minha mãe

Rosalina A Mendes dos Santos

e do meu sobrinho

Jósimo dos Santos Barbosa

AGRADECIMENTOS

A Deus pelo o dom da vida, pela misericórdia, amor e proteção diária.

Ao meu pai Antônio Bezerra dos Santos pelas lições de vida e por me ensinar a lutar sempre, sem jamais desistir. Além, dos ensinamentos empíricos do valor e da força não só do trabalho mais dos próprios fundamentos da determinação humana.

As minhas irmãs Lúcia, Lucimar, Lucilene e irmãos Jaime e Jair pela atenção, carinho, apoio e referência dados para as minhas buscas no decorrer desse trabalho, principalmente, nos momentos de angústia e solidão, apesar de toda à distância geográfica que nos separam neste momento.

Aos meus *amorzinhos*, minhas sobrinhas, Mônica, Macela, Luíza, Vitória, Denise, Ana Cláudia, Ana Karolina e Érika e meus sobrinhos Leonardo e Adriano, pelo carinho das boas lembranças do *tio flankiko*.

A Ana Santos, companheira que dividiu comigo os momentos difíceis e alegres dessa travessia. Por toda sua dedicação meu MUITO OBRIGADO.

A minha sogra Lucy Castro por todo auxílio e afeição a mim conferido, meu MUITO OBRIGADO.

A minha cunhada Suely Castro pela ajuda indispensável nos momentos cruciais.

Aos meus ex-alunos e agora colegas de trabalho, no CEFET-Am, Emmerson Santa Rita e Sérgio Coelho por suas indispensáveis contribuições.

Aos meus mestres orientadores, Prof^o João Bosco Dr. e Prof^o Ilson Wilmar Dr., pelo incentivo, amizade, orientação e principalmente pelas referências de honestidade, dentre outras, com que me brindaram durante todo o curso.

Ao colega João Carlos Linhares por toda colaboração dispensada nesse trabalho.

As funcionárias mais dedicadas da Pós-Graduação da Ciência da Computação, Valdete e Verinha, pelas orientações, esclarecimentos, encaminhamentos e principalmente pela paciência e eficiência no atendimento às solicitações feitas.

A família Thomé, Zeina, Lauro (*in memoriam*) e Júlio, pela amizade, carinho, dedicação e por todos os encontros Amazônicos.

Aos nossos amigos Norberto, Clarice Etges e Araci Hack Catapan pelo carinho, amizade e assistência indispensáveis para a realização desse curso.

E a todos que direta ou indiretamente contribuíram para a realização desse curso e dissertação, meu MUITO OBRIGADO!

SUMÁRIO

LISTA DE FIGURAS	ix
LISTA DE TABELAS	xi
RESUMO	xii
ABSTRACT	xiii
1 INTRODUÇÃO	
1.1 Apresentação	14
1.2 Problematização	15
1.3 Justificativa	22
1.4 Objetivos	
1.4.1 Geral	24
1.4.2 Específicos	24
1.5 Metodologia	25
1.6 Estrutura do trabalho	26
2 FUNDAMENTAÇÃO TEÓRICA	
2.1 Tecnologia <i>Design by feature</i>	27
2.1.1 Definições de <i>feature</i>	27
2.1.2 Modelamento de peça baseada em <i>feature</i>	29
2.1.3 Pesquisas sobre features	30
2.2 Árvore de Estrutura hierárquica de funções de peça	31
2.2.1 Estrutura funcional de peça	32
2.2.2 Estrutura conceitual de peça	33
2.2.3 Exemplo de aplicação da árvore hierárquica de funções de peça	34
2.3 Estado da arte	
2.3.1 Processamento da Linguagem Natural - PLN	37
2.3.1.1 Alguns conceitos em lingüística	41
a) Frase como estrutura	42
b) Definição de Frase	43

c) Princípios de organização da estrutura frasal	44
d) Diferentes processos de segmentação	44
e) Sintagmas como constituintes operacionais	45
f) Conceito e classificação de sintagma	45
2.3.2 Conhecimento e linguagem	47
2.3.3 Etapas do Processamento de Linguagem Natural	48
2.3.3.1 Processamento Morfológico	48
2.3.3.2 Processamento Sintático	49
2.3.3.3 Processamento Semântico	50
2.3.3.4 Processamento do Discurso	52
2.3.3.5 Processamento Pragmático	52
2.3.4 Gramática e Parsing	53
2.3.5 Gramática formal	53
 3 Modelo de <i>Parser</i> para o processo de projeto em ambiente de sistema mecânico	
3.1 Hierarquias: definição, níveis e complexidade	57
3.2 Sentença como estrutura hierárquica complexa	58
3.2.1 Definição de estrutura sentencial	58
3.2.2 Relações entre os elementos de uma estrutura hierarquia	60
3.3 Arquitetura do modelo de <i>PARSER</i> proposto	67
3.3.1 Módulo Segmentador de Texto	67
3.3.2 Módulo Tokenizador de Sentença	69
3.3.3 Módulo Léxico	71
3.3.4 Módulo Analisador Léxico/Morfológico	72
3.3.5 Módulo Composição de Termos Compostos	72
3.3.6 Módulo Analisador Sintático	72
3.3.7 Módulo Analisador Semântico	75

3.3.7.1	Sistema de Codificação & Classificação de peça KK-3	77
4	Conclusões e Trabalhos Futuros	
4.1	Conclusões	80
4.2	Trabalhos Futuros	81
	REFERÊNCIAS BIBLIOGRÁFICAS	83
	APÊNDICE A – Trabalhos parciais decorrentes do processo de Mestrado	86
	APÊNDICE B – Algumas árvores funcionais das peças mecânicas analisadas	87

LISTA DE FIGURAS

Figura 01: Estrutura Geral do Processo de Projeto	15
Figura 02: Localização do trabalho na estrutura do processo geral de projeto	17
Figura 03: Biela X Necessidade de soluções volumétricas	19
Figura 04: Biela X Soluções <i>Featurizadas</i>	20
Figura 05: Estrutura funcional de peça, (Apud Linhares)	32
Figura 06: Estrutura conceitual de peça, (Apud Linhares)	34
Figura 07: Estrutura hierárquica de peça, (Apud Linhares)	34
Figura 08: Correlação entre a estrutura hierárquica de funções, ordem dos níveis funcionais e categoria das funcionalidades de peça	35
Figura 09: Desdobramento da função global de peça ‘Transformar movimento orbital em movimento alternativo’	36
Figura 10: Árvore hierárquica com suas respectivas <i>featurizações</i>	37
Figura 11: Árvore sintática: ‘Preciso de biela de ligação para excêntricos’	50
Figura 12: Esquema em bloco do <i>Parser</i> :	52
Figura 13: Uma gramática simples para uma linguagem natural	55
Figura 14: Hierarquia de Chomsky	56
Figura 15: Elemento membro de uma Estrutura E	61
Figura 16: Primeiro elemento de uma Estrutura E	61
Figura 17: Elementos Y como sucessor de X na Estrutura E	61
Figura 18: Z como Elemento antecessor de W na Estrutura E	61
Figura 19: Elemento X à direita da subEstrutura S_E na Estrutura E	62
Figura 20: Elemento X à esquerda da subEstrutura S_E na Estrutura E	62
Figura 21: Último elemento da Estrutura E	62
Figura 22: Elemento com ordem crescente na Estrutura	63
Figura 23: Elemento com ordem decrescente na Estrutura	63
Figura 24: Concatenação dos elementos de Estruturas diferentes em uma única Estrutura E	63

Figura 25: Substituição de um ou mais elementos na Estrutura E	64
Figura 26: Retirada de elementos repetidos em uma Estrutura	64
Figura 27: Operação de Complementaridade entre Estrutura subEstrutura	65
Figura 28: Operação de Complementaridade entre Estruturas	65
Figura 29: Agrupamento de elementos em uma única Estrutura	65
Figura 30: Desagrupamento dos elementos de uma Estrutura	66
Figura 31: Arquitetura do Modelo de <i>PARSER</i> . Proposto	67
Figura 32: Resultado da segmentação do texto biela tipo 14	68
Figura 33: Protótipo com os módulos opcionais como Segmentador, Tokenizador, Verificador de Termos Compostos - Versão 1.0	70
Figura 34: Protótipo com os módulos opcionais de pesquisa de peças mecânicas através do Nome da Peça, FGp/FPp/FEp ou Árvore Funcional via texto – Versão 1.0	76
Figura 35: Processamento da árvore de funcionalidades da peça biela tipo 01 via código KK-3	78
Figura 36: Informações sobre a interface do bando de peças mecânicas	79
Figura 37: Interface do banco de peças mecânicas	79

LISTA DE TABELAS

Tabela 01: Características da linguagem que a tornam tão difícil quanto útil, compiladas de Rich & Knight	39
Tabela 02: Características dos elementos formadores de cada nível	59

RESUMO

Este trabalho tem como objetivo propor um modelo para implementação de um *Parser* para ambientes de projeto de produtos com ênfase no projeto de peças, componentes ou sistemas mecânicos. Trata-se de uma proposta de auxílio computacional estruturada a partir do processamento da linguagem natural dos projetistas desses ambientes, baseada na análise da estrutura funcional da peça segundo a sua função global ou as suas funções parciais ou elementares. O modelo pretende analisar e verificar a possibilidade de *featurização* da intenção morfológica do projetista, o que poderia viabilizar a integração do próprio processo de projeto de peças, componentes ou sistemas mecânicos com sistemas CAD, tendo como suporte bibliotecas ou de bancos de dados de *features* de domínio específico. A perspectiva desse trabalho é a da superação da abordagem de procedimentos que sugerem o uso excessivo e direcionador de questionários e, principalmente, da limitação da linguagem escrita dos projetistas. Na estrutura do processo de projeto, este trabalho se situa entre as fases de projeto conceitual e preliminar, onde as intenções morfológicas do projetista, em relação às peças, podem ser identificadas, estruturadas e materializadas para o produto, tornando-se um aspecto central e explícito do mesmo.

Palavras-chave: Processamento da Linguagem Natural (PLN); funções da peça (FGp, FPP e FEp); tecnologia *Design by feature*; Intenção Morfológica do projetista.

ABSTRACT

This work aims to propose a model to implement a parser for environments of product projection, with emphasis on the projection of parts, components or mechanical systems. It is a proposal for computational aid, structured on the processing of the natural language used by the projectionists of these environments, based on the analysis of the functional structure of the part, according to its global function or its partial *or* elemental functions. The model aims to analyse and verify the possibility of 'featurization' of the projectionist's morphological intention, which could lead to the integration of the process of parts, components or mechanical system projection in itself with the CAD system, with the support of libraries or data banks of features of specific context. The present work also aims to offer a better alternative than the procedure approach, which proposes an excessive and restrictive use of questionnaires and, above all, a better solution than the limiting written language used by projectionists. Within the structure of project processing, this work is located between the steps of conceptual and preliminary project, where the projectionist's morphological intentions regarding the parts can be identified, structured and materialised in the product, becoming a central and explicit aspect of it.

Key words: natural language processing (PLN); functions of the part the projectionist's (GFp, PFp and Efp); *design by feature* technology; morphological intention.

1 INTRODUÇÃO

1.1 Apresentação

O fenômeno da globalização, a contínua corrida por baixos custos, a diversificação da produtividade e de matérias-primas, a complexificação¹ do processo de industrialização a partir das ferramentas informáticas, já são velhas conhecidas das empresas. Busca-se hoje, cada vez mais, a integração de ferramentas, equipamentos, máquinas, softwares, linguagens e sistemas capazes de tornar ainda mais eficiente o próprio processo produtivo, aumentando não só a qualidade, mas, principalmente, otimizando custos. Nesse sentido, algumas áreas do conhecimento propõem-se dotar as máquinas de um certo racionalismo, de metodologias, técnicas, celeridade, heurística e expansão da memória, para que as mesmas possam servir cada vez mais de suporte ao pensamento e ampliar as capacidades humanas.

É diante desse contexto que o projeto de sistemas vem se complexificando. São muitas as pesquisas que vêm *pari passo* sistematizando e informatizando as fases iniciais do processo de projeto. Pode-se citar, por exemplo, aquelas que envolvem a sistematização da fase informacional de Fonseca (2000) com o **SEPI** (Sistema de Apoio à Obtenção das Especificações de Projeto de Produtos Industriais). Fonseca aponta outros estudos como a proposta de sistematização da fase conceitual de Ogliari (1999) com o **DEFENEC** (Programa de Apoio à Definição de Necessidades de Projeto) e **QFD** (Programa da “Casa da Qualidade”), o **WINSAPPI** (Sistema de Apoio ao Projeto de Produtos Industriais) de Silva (1995), o **SADEPRO** (Sistema Auxiliado por Computador para Desenvolvimento de Produtos Industriais) de Fiod (1993), dentre outros.

Este trabalho situa-se entre as fases de projeto conceitual e preliminar do produto, pois é nesse momento que se estabelece um conjunto de requisitos e restrições, a saber, as especificações físicas. As referidas especificações proporcionaram a determinação dos princípios de soluções para a resolução do problema de projeto. Assim sendo, esses princípios de soluções do produto também possibilitaram a estruturação das

¹ **Complexificação** é um termo utilizado por Pierre Lévy notadamente na sua obra denominada de *Tecnologias da Inteligência* (1997). Reflete a co-existência de modos fundamentais de gestão social do conhecimento como a oralidade, a escrita e a informática. Relaciona-se ainda, a técnicas de armazenamento e processamento das representações que tornam possíveis evoluções culturais e novas formas de construções do devir coletivo, permeadas pela heterogeneidade de hardwares, sistemas, softwares, linguagens, dentre outras.

funcionalidades de cada peça e módulo realizável (MR). Entende-se por MR, um conjunto harmônico de peças que exercem uma função específica no produto.

Logo, partindo-se do pressuposto de que qualquer componente, peça ou módulo realizável pode ser estruturado em uma hierarquia de funcionalidades, então, o problema consiste em implementar um processador da linguagem natural técnica. Nessa perspectiva, esse trabalho se constitui como uma das principais ferramentas possibilitadoras de sistematização da própria estrutura do processo de projeto de produto.

1.2 Problematização

A estrutura do processo de projeto mais bem aceita na comunidade científica nacional ou internacional, é a proposta metodológica de Pahl & Beitz (1996: 68). A fase denominada inicialmente pelos referidos autores como esclarecimento ou definição da tarefa de projeto, foi redimensionada por Fonseca (2000: 32) passando a ser chamada de projeto informacional conforme apresentado na Figura 01. Baseando-se nessa estrutura, este trabalho se situa na transição das fases de projeto conceitual e preliminar ou na 4ª fase da VDI 2221, estruturação em módulos realizáveis (MR's).

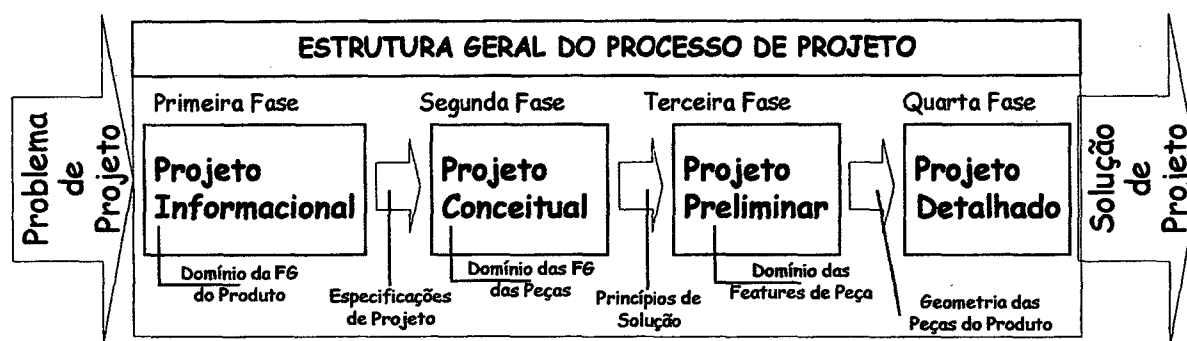


Figura 01 – Estrutura Geral do Processo de Projeto

A estrutura do processo de projeto mencionada acima se configura da seguinte maneira:

Na primeira fase, a do projeto informacional, são colhidas informações que servirão de base para a execução das suas etapas subseqüentes. Segundo Fonseca (2000), trata-se de informações que dizem respeito a quatro categorias, a saber:

- i. necessidades – são externas à equipe de projeto e devem ser identificadas junto àqueles clientes e usuários relacionados com o produto a ser concebido. As necessidades constituem-se na base para se chegar às especificações de projeto, as quais configurarão a principal função do produto. No dizer de Fonseca (2000, p. 41)
... "as necessidades externas à equipe de projeto, podem ser entendidas de maneiras diversas, possibilitando induzir opiniões diferentes, ou ainda, de forma confusa. Isto significa que toda necessidade bruta (seja na forma escrita ou falada) deve ser submetida a uma análise de significado. Isso implica numa procura por consenso da equipe de projeto sobre o significado do que se expõe na necessidade".
- ii. requisitos de usuário – são os critérios decorrentes da análise de significado, que padronizarão a solução pretendida. Essa solução deve suprir as necessidades apresentadas pelos clientes;
- iii. requisitos de projeto – defini-los consiste na etapa inicial e fundamental para a elaboração de qualquer projeto em engenharia. “São os aspectos físicos e técnicos mensuráveis do produto (...)” (Fonseca, 2000: 44). Esses aspectos podem ser descritos através de características técnicas, possíveis de serem mensuradas por algum tipo de conversão, como por exemplo, pela casa da qualidade²;
- iv. especificações de projeto – as especificações de projeto vão estabelecer as metas, objetivos e restrições que servirão como guia para a definição da função global do produto a partir dos requisitos de projeto.

A segunda fase, a do projeto conceitual, pauta-se nas proposições de PAHL & BEITZ (1996). Segundo essas proposições, o projeto conceitual inicia com o estabelecimento dos problemas essenciais de projeto, partindo das especificações de projeto ou lista de requisitos, na forma de funções do produto. Encerra com o desenvolvimento e avaliação dos melhores arranjos de MR e peças, que correspondem às soluções conceituais também chamadas de princípios de solução para o produto.

Esses conceitos, que também são transformações de informações como as da fase de projeto informacional, representam o produto em suas principais funcionalidades e princípios de solução. São concepções de natureza qualitativa, elaborações de processos (ou subprocessos) que têm como propósito obter uma ou mais concepções para o produto as quais se caracterizam através de esquemas ou esboços de soluções, desenvolvidas e

² Consiste em um mapa conceitual que permite um planejamento interfuncional e comunicativo entre os setores responsáveis pelo desenvolvimento do produto em todas as suas etapas.

organizadas numa dada estrutura. As funções resultantes desse processo irão orientar a busca, a sistematização e o arranjo de princípios de solução alternativos para o produto nas etapas posteriores.

Nessa fase são desenvolvidos ainda, princípios de solução para identificar problemas essenciais, estabelecer estruturas de função, pesquisar princípios e estruturas de operação, combinar e reconhecer as variantes de concepção e finalmente avaliar contra-crêterios técnicos e econômicos. Conforme assinalado por Linhares *“após o projeto conceitual do produto, deve-se executar o projeto conceitual específico de suas peças cujas informações podem ser representadas de forma estruturada através de atributos e operações visando a disponibilização destas informações ao processo de projeto de produto como um todo.”* (Cf. 2000, p. 34)

Sabe-se que, no espaço entre as fases do projeto conceitual e preliminar, os projetistas elaboram esquemas (ainda que no plano das idéias) não definitivos. Esses esquemas consistem em tentativas de configuração dos subsistemas que conformam o produto como um todo e conseqüentemente cada uma de suas peças ou módulos realizáveis. Uma forma de se visualizar, ainda que virtualmente esse processo, é emulando a própria construção de cada peça. Isso é possível através do uso de sistema CAD a partir das estruturas funcional e conceitual das mesmas. (Ver Figura 02).

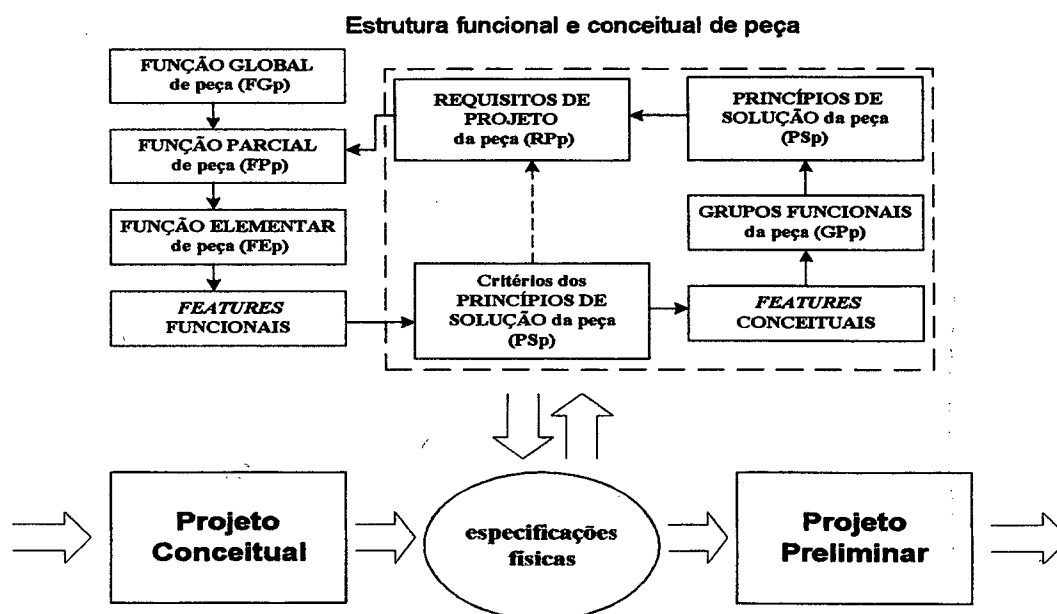


Figura 02: Localização do trabalho na estrutura do processo geral de projeto

Na terceira fase, a do projeto preliminar, sobre a melhor concepção – de acordo com os critérios de seleção adequados aos contextos de projeto de produto - desenvolvem-se processos para configurar o layout do produto. Esse layout, de natureza quantitativa, consiste no arranjo geral dos elementos que caracterizam o produto em suas principais geometrias e formas. Determinam-se, assim, as dimensões básicas das peças que comporão o produto, considerando a dinâmica das peças e as principais propriedades de seus materiais. Salienta-se que nessa fase define-se a estrutura conceitual e funcional de cada peça, bem como todos os MR's do produto. Essas estruturas são definidas primeiro devido à necessidade de se estabelecer o relacionamento das interfaces entre as diversas peças. Segundo, por ser imprescindível não só selecionar os melhores layouts preliminares, mais ainda refinar e depurar cada um deles.

Na quarta fase, a do projeto detalhado, efetuam-se processos requeridos para transformar cada layout do produto, MR e/ou peça, em documentos que caracterizam detalhadamente as soluções desenvolvidas e que possibilitam a sua realização física. Possibilitam também, o estabelecimento de dimensionamentos e cálculos de resistência, confiabilidade, enfim, de todos os atributos do ciclo de vida do produto que são prioritários. Nesse momento, as decisões tomadas acerca de dimensões, arranjos, formas dos componentes individuais, acabamentos superficiais, materiais e custos de fabricação são tratados, possibilitando a preparação da documentação final do produto.

Sabe-se que na fase de projeto detalhado, há sistemas informáticos mais capazes de detalhar através de *features* geométricas cada uma das peças do projeto do produto, como é o caso, por exemplo, de sistemas CAD como *Unigraphics*, *MicroStation*, *Solidedge*, *SolidWorks*, *CATIA*, *Pro-Engineer*. Portanto, as informações geométricas (volumétricas ou espaciais ou até mesmo planares) e não-geométricas (materiais, ergonomia, manufatura, montagem, função do produto, operação, manutenção e custos) podem estar dispostas nos MR's, conjunto de sistema técnico, e ainda em cada componente ou peça.

Dessa forma, a função principal de cada peça deve proporcionar a visão global de sua topologia e também de suas interfaces com as demais peças do mesmo conjunto. Em se tratando de profissionais versados em projeto de peças, componentes e sistemas mecânicos, ao perceber a função principal da peça, conseguem visualizar mentalmente uma série de modelos cumpridores dessa função. As descrições da intenção do projetista

nesse processo de projeto polarizam formas geométricas, de modo a realizar os esquemas relativos a cada um dos modelos imaginados, incorporando sua geometria correspondente.

Portanto, a polarização de uma forma geométrica nada mais é que a tentativa de *featurizar*³ uma descrição da intenção do projetista numa forma, a priori que seja, planar, volumétrica ou espacial e que satisfaça parcial ou completamente essa intencionalidade, atendendo necessariamente os próprios requisitos do projeto ou reprojeção da peça. Nessa perspectiva, a intencionalidade do projetista é guiada por sua experiência, catálogos de soluções (ex. *TRIZ* - sigla russa para Teoria da Solução Inventiva de Problemas) ou outros tipos de solução.

O exemplo das Figuras 03 e 04 abaixo, nas amostras (a) e (b) de cada caso, mostram as polarizações das formas geométricas volumétricas nas *extremidades da biela* de um compressor alternativo, em função de uma solução para melhor suportar esforços de tensões na transformação de movimento rotacional em alternativo. Para tal, o projetista criou a partir do projeto original uma nova peça através da definição de seis (06) *features* distintas como mostrada na figura 4(b).

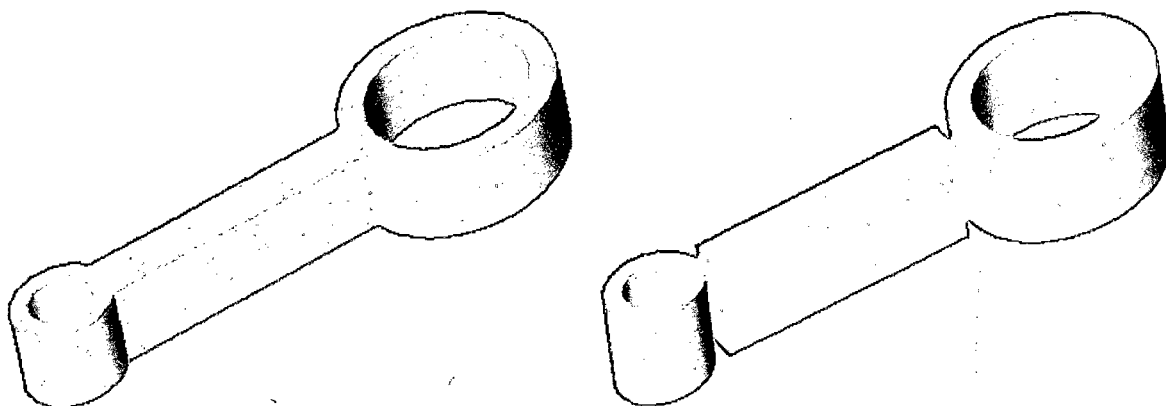
Observe as diferenças.

Figura 03: Biela X Necessidade de soluções volumétricas

(a) Biela

(b) Biela discretizada

(com problema de concepção no cumprimento da sua função)



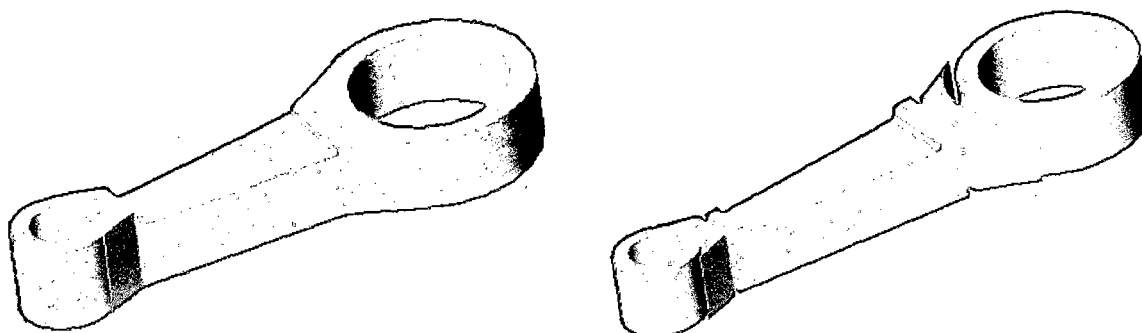
³ *Featurizar* consiste na capacidade de transformação de uma intenção, vontade, necessidade, funcionalidade, enfim, de uma abstração em uma forma geométrica planar, volumétrica ou espacial possível de realização.

Figura 04: Biela X Soluções *Featurizadas*

(a) Biela

(b) Biela discretizada

(com as intenções de soluções *featurizadas*)



Nesta fase de projeto detalhado, as formas geométricas de cada uma das peças do produto que antes permaneciam no plano das idéias, passam a ser inscritas através de documentos. Esses, por sua vez, ficam disponibilizados para completarem o ciclo de vida do produto.

Então, finalmente nesta fase, preparam-se documentos de produção e operação para se elaborar cada *sketch* detalhadamente e listas de peças, completar as instruções de produção, montagem, transporte e operação e verificar documentos complementares.

No escopo dessa dissertação, as informações encapsuladas na linguagem escrita dos profissionais de projeto de componente, peça e sistema mecânico, bem como nas descrições próprias de sua linguagem coloquial, serão analisadas a fim de se identificar às regularidades da gramática geradora, empreendedora e/ou provedora de movimento e ação. Outras regularidades, como aquelas que especificam detalhes de complementaridade do objeto da ação, constituir-se-ão em importantes filtros das informações concedidas, no nosso entendimento, por projetistas ainda mais hábeis e com conhecimentos tácitos⁴. Por isso da pretensa idéia de identificar os fenômenos sintáticos descritos na linguagem escrita dos projetistas, pois é nessa estrutura frasal que eles articulam toda a bagagem semântica adquirida no dia-a-dia. Então, ao se analisar cada uma das sentenças usadas por especialistas dessa área para expressar as funcionalidades de peça, espera-se poder detectar todo o padrão possível desse estilo frasal em uso e com isso delimitar seu relacionamento

⁴ **Conhecimento tácito** de acordo com Nonaka e Takeuchi (1997) é aquele que se obtém pela experiência e que só se comunica indiretamente por metáforas e analogias.

léxico/morfológico, sua organização sintática, como também as propriedades e tipos frasais.

Segundo Chomsky (apud Nivette: 1975, 43) “a gramática gerativa transformacional compreende três componentes: o sintático, o semântico e o fonológico. O primeiro é o responsável pela geração de todas as frases da língua, o segundo traduz a estrutura sintática subjacente em termos de uma representação semântica e o terceiro atribui uma representação fonética à estrutura superficial”. E é a partir das análises das representações semânticas das descrições supracitadas que se pretende alcançar um outro objetivo relacionado ao modelo proposto, a saber: analisar a possibilidade de *featurização* da intenção morfológica do projetista, viabilizando assim, a integração do processo de projeto de sistemas mecânicos a sistemas CAD, via uma linguagem simbólica.

Interpretar a linguagem natural dos profissionais de sistemas mecânicos significa empreender ações que visem transformar sua estrutura frasal coloquial numa outra, a da linguagem simbólica, adequada para ser correlacionada com linguagens de sistemas CAD. O objetivo dessa transformação informacional, é capturar e manter a intenção na comunicação do projetista, auxiliando-o na sua liberdade criativa⁵ como profissional de sistemas mecânicos.

Isto posto, a intenção da qual falamos é a intenção intrínseca de Searle (apud Teixeira: 2000, p. 138), entendida como a que está

... “presente no discurso lingüístico, e constitui uma forma derivada de intencionalidade que consiste na relação das representações lingüísticas com os estados intencionais, o que permite que estas últimas sejam representações de alguma coisa do ‘contexto do projetista de sistemas mecânicos’”. (grifo nosso)

Depreende-se pois, que sem essa relação entre representações mentais ou estados intencionais e representações lingüísticas não podemos falar de compreensão de frases ou compreensão lingüística, visto que as relações entre as representações lingüísticas e os estados intencionais transformam o código lingüístico num conjunto de signos, ou seja, estabelece o seu significado. Trata-se, no dizer de Searle, da intencionalidade intrínseca que constitui a condição necessária para que um sistema simbólico adquira uma dimensão semântica. Sem ela [a dimensão semântica], não podemos falar de compreensão.

⁵ *Criatividade* é a habilidade do especialista (projetista) de ter idéias novas e úteis para resolver o problema proposto ou sugerir soluções para a concepção de um produto (Back & Forcellini: 1998, 4-1).

1.3 Justificativa

Dada a crescente competitividade das empresas existentes seja no mercado nacional ou internacional, o uso sistemático de tecnologia informática nas atividades de concepção de produtos, em processos de projeto de sistemas técnicos (principalmente) nas fases iniciais tem sido ultimamente, uma tônica na Engenharia Mecânica.

No entanto, existem dificuldades próprias do contexto da concepção de projeto de sistemas mecânicos oriundas de sua estruturação sistêmica. Tais concepções, dependem da experiência do projetista em outros projetos de natureza semelhante, de analogias com sistemas existentes, de sua intuição (devido ao alto grau de abstração nas fases iniciais), da resolução pelo método das tentativas e erros, do seu julgamento próprio e singular, de pesquisas exaustivas. Muitas são as metodologias de projeto de concepção de produto existentes atualmente. Entretanto, dada às vantagens das ferramentas informática, existe uma preferência por aquelas que se propõem a se integrar com sistema CAD, principalmente aquelas que agregam os fundamentos conceituais do próprio processo de projeto através de aspectos de implementação computacional. Isso porque o uso de tecnologia possibilitadora do aumento das capacidades de resolução do “Problema de projeto” beneficiaria o projetista não só através de suas funcionalidades, velocidade e capacidade do processamento numérico, mas também, traria implícito nos seus processos a própria metodologia de projeto, além de outros métodos de sistematização, obrigando não só os projetistas, mas a todos da área de processo de projeto a segui-las.

Adicionalmente, tecnologias dessa natureza prestam-se muito bem para atividades de ensino e treinamento em metodologias de projeto, melhorando o aprendizado dos engenheiros e futuros engenheiros projetistas estimulando o desenvolvimento sistematizado de produtos.

Nessa perspectiva, este trabalho propõe-se realizar a implementação de um Modelo de *PARSER* para auxiliar o projetista no processo de concepção de projeto de componentes, peças e sistemas mecânicos. As informações aqui analisadas, a partir da representação do conhecimento extraído do especialista, serão as representadas através de sentenças, a função global e as funções parciais e elementares. Essas funções são as comumente usadas no ambiente de trabalho desses profissionais, descrições próprias de sua linguagem coloquial.

Aqui a idéia é capturar, não só, o conhecimento explícito do projetista no seu *savoir-faire*, aquele conhecimento que pode ser articulado na linguagem formal, inclusive em afirmações gramaticais, especificações, manuais e assim por diante. Acredita-se que este tipo de conhecimento pode ser então transmitido, formal e facilmente para os indivíduos desse ambiente.

Propõe-se ainda, compor um *CORPUS* da linguagem dos profissionais de projeto de sistemas mecânicos, componentes, peças ou sistemas mecânicos do produto compressores alternativos. O referido *CORPUS* tem duas finalidades. A primeira diz respeito à criação de uma metodologia para se aprimorar o *PARSER*. A segunda busca por elementos para se experienciar o que Nonaka & Takeuchi (1997, p. XIII), em anos de pesquisa sobre as empresas japonesas concluíram: “a criação do conhecimento nas empresas é a principal fonte de competitividade internacional”.

Assim, interpretar a linguagem natural desses profissionais de sistemas técnicos implica em transformar sua estrutura frasal numa outra, da linguagem de projeto, adequada para ser correlacionada e até mesmo integrada pela linguagem simbólica computacional (por exemplo: as bibliotecas de CAD, como as de *features*). Essa transformação informacional visa capturar e manter a intenção na comunicação do projetista, auxiliando-o na sua liberdade criativa⁶ como profissional de sistemas mecânicos. Constitui-se em mais um desafio aqui proposto o de propiciar ao projetista uma certa celeridade no seu fazer como engenheiro projetista de sistemas mecânicos.

Antes de tudo porém, compartilha-se do alerta Mito et al (2000, p. 14), longe de se querer construir uma teoria científica da organização sintática das sentenças, faz-se necessário, mesmo assim, observar as estruturas sintáticas das sentenças que efetivamente são próprias da língua daqueles profissionais, sem ignorar nenhuma delas. Ressalta-se ainda, que o interesse por uma formulação de princípios existentes na base dessas organizações sintáticas é de extrema importância para o sucesso desse trabalho.

Assim sendo, espera-se que a utilização de categorias e conceitos gramaticais presentes direta ou indiretamente na produção dessa linguagem técnica em ambiente de projeto de sistemas mecânicos, se legitime na sua forma singular de existência.

⁶ Segundo Back & Forcellini (1998,4-1).

1.4 Objetivos Do Trabalho

1.4.1 Objetivo Geral

Implementar um Modelo de *PARSER* para auxiliar o projetista de projetos de peça, componente ou sistema mecânico na sua concepção de projeto, capaz de processar a linguagem natural dos profissionais desses ambientes, a partir da *função global e das funções parciais e elementares*.

1.4.2 Objetivos Específicos

- a) Compor um *CORPUS* da linguagem dos profissionais de projeto de sistemas mecânicos, componentes e peças de compressores alternativos;
- b) Analisar a estrutura lingüística do especialista de projetos de peças para produtos industriais, identificando os fenômenos sintáticos descritos nessas estruturas (como, por exemplo, o relacionamento léxico/morfológico, a organização oracional, suas propriedades e influência) na determinação dos tipos frasais dos atores envolvidos;
- c) Identificar as correlações das estruturas lingüísticas do especialista e metodologias de projetos de peças, componentes e sistemas mecânicos industriais, de acordo com a categoria do contexto do sistema mecânico de domínio;
- d) Propor a arquitetura de um *PARSER* para processar a linguagem dos profissionais de projetos de peça, componente ou sistemas mecânicos nas suas concepções cujas entradas são as funcionalidades de projeto;
- e) Testar o Modelo de *PARSER* protótipo.

1.5 Metodologia

Esta dissertação se propõe a captar com todo o detalhamento e rigor, os fundamentos da sistematização dos processos de projeto de produto industrial a partir da estrutura geral de projeto, analisando suas fases e estabelecendo suas possíveis inter-relações. Só depois de cumprida essa tarefa, ter-se-á elementos para expor adequadamente como os profissionais desses ambientes encapsulam suas intenções morfológicas, através da sua linguagem coloquial, para resolver os problemas no processo de projeto de um produto.

As fases desta pesquisa evoluem de forma teórica e prática em duas dimensões que se constituem como um processo inteiramente articulado. Realizam-se investigações bibliográficas e de campo para verificar os vínculos e as inter-relações entre o processo de projeto de peças, componentes e sistemas mecânicos e o projeto do produto. Nesses termos, busca-se uma maneira de auxiliá-lo a determinar os elementos funcionais e conceituais que integram as alternativas de solução e as configurações de peça que alimentam a geração das alternativas de solução que o compõem.

Nas pesquisas bibliográficas da estrutura geral do processo de projeto de produto, busca-se conhecer melhor não só o próprio processo, mas também, apreender a evolução das técnicas do processo de sistematização dessa estrutura. Adicionalmente, faz-se uso de uma metodologia que visualize o processo de projeto do produto de forma estruturada, pois, dessa forma, ao utilizar-se as ferramentas informáticas pode-se mediar o processo de projeto de produto de modo a usar as principais vantagens dessas tecnologias, que são a velocidade de processamento e a capacidade de memória.

Assim sendo, espera-se associar as tecnologias informáticas ao processo de projeto de produto, a partir de um sistema de processamento de linguagem natural. Esse sistema teria a função de transformar a linguagem natural dos profissionais desse ambiente, em linguagem digital e logo após, numa linguagem simbólica o que seria denominado como o próprio código da peça. Dessa forma, a linguagem simbólica poderia ser correlacionada com a peça para o projeto em questão, a partir de um banco de dados.

A investigação de campo se dá com o propósito de compor um *CORPUS* com as estruturas sentenciais da linguagem coloquial dos projetistas de peça, componentes e sistemas mecânicos. No contexto do referido sistema, busca-se descrever, interpretar e

explicar o resultado coletado, de sorte que venha a fundamentar respostas às indagações, visando estabelecer relações entre os dados obtidos e as hipóteses formuladas na problematização. Na interpretação dos dados, procurar-se-á encontrar o significado mais amplo das respostas, em relação aos objetivos e às indagações do proposto no escopo deste trabalho, tendo como demarcador as categorias selecionadas para este estudo.

Finalmente, espera-se implementar o Modelo do *PARSER* para o processo da linguagem técnica dos profissionais da área de projeto de sistemas mecânicos, de sorte que após uma avaliação do sistema informático - através de testes do mesmo - esse possa comportar-se de forma adequada para os fins aqui propostos.

1.6 Estrutura da Dissertação

Além deste capítulo introdutório, este trabalho está organizado conforme os outros elencados a seguir:

O Capítulo II apresenta a fundamentação teórica referente à Tecnologia “*Design by feature*” e o Processamento da Linguagem Natural (PLN);

No terceiro capítulo:

- detalha-se o plano metodológico da criação das regras de produção, cujos estudos convergiram na aplicação das mesmas quando da determinação de tipos de frases e de todas as estruturas usadas na implementação do Modelo de *PARSER*;
- propõe-se a arquitetura do Modelo de *Parser* para projeto de peças, componentes e sistemas mecânicos, bem como a descrição funcional de cada um de seus módulos;

No quarto capítulo, apresentam-se as conclusões e recomendações para trabalhos futuros;

Na parte final têm-se as Referências Bibliográficas e os Anexos, listando os trabalhos decorrentes do processo de mestrado e algumas árvores funcionais das peças mecânicas usadas na concepção e no desenvolvimento desta dissertação.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Tecnologia *Design by feature*

A utilização da tecnologia *Design by feature* como base de informação para a modelagem do produto é uma das mais novas perspectivas voltada para a sistematização das atividades de desenvolvimento de um novo produto. De acordo com Salomons (1993:113), essa tecnologia é o caminho mais adequado para se promover a integração entre as atividades de projeto, planejamento de processos, fabricação, inspeção, dentre outras.

Essa perspectiva é confirmada por outras atividades de desenvolvimentos de produto, como mostrado em Rosa (s/d: 2)

“O estudo voltado para a sistematização das atividades de desenvolvimento de um novo produto, realizado nos últimos anos, tem permitido vislumbrar as técnicas e estratégias a serem adotadas, surgindo os conceitos de ‘design for manufacturing’, ‘design for assembly’, ‘concurrent engineering’, ‘life cycle design’, ‘Design by feature’ e muitos outros”.

2.1.1 Definições de *feature*

Segundo investigações bibliográficas, as pesquisas voltadas para a tecnologia *Design by feature* são relativamente recentes e várias são as definições encontradas, cada uma elaborada com base em conceitos da sua área específica. Segue-se abaixo, algumas definições encontradas na literatura:

➤ Shah, Rogers et al. (1990:233) apresentam o conceito de *feature* de forma a partir da identificação de alguns atributos da sua própria geometria;

➤ Mayer et al. (1994:49) apresenta várias definições de *feature*, cada uma aplicada a uma área distinta:

☞ *Feature* de forma: entidades relacionadas com a geometria e topologia de uma peça;

☞ *Feature* de tolerância: entidade relacionada com os desvios aceitáveis nas dimensões de uma peça;

☞ *Feature* de material: entidade relacionada com a funcionalidade de peça;

☞ *Feature* de montagem: entidade relacionada às operações de montagem;

☛ *Feature* é uma forma geométrica definida por um conjunto de parâmetros que têm significado especial para engenheiros de projeto e fabricação;

Irani et al. (1995:21) define *feature* do ponto de vista de planejamento de processo. Assim, uma *feature* pode ser identificada como uma modificação na forma, no acabamento superficial ou nas dimensões de uma peça, produzida por um determinado conjunto de operações;

Salomons et al. (1993:115) em sua revisão sobre pesquisas com projeto baseado nas tecnologias *Design by feature* apresentam uma série de definições, algumas das quais foram coletadas na literatura. Sendo assim, essas tecnologias podem ser entendidas como:

- Um conjunto de informações referentes à forma, assim como outros atributos de uma peça, de tal modo que, essas informações possam ser manipuladas no projeto de manufatura e montagem (definição apresentada pelos autores);
- Configurações geométricas específicas formadas em uma superfície, aresta ou canto de uma peça. Este conceito é elaborado tendo por base o ponto de vista de planejamento de processos (Glossário Ilustrado de *Features* da CAM-I);
- Uma forma genérica que tem algum significado em engenharia (Wingard);
- Um conjunto de informações usadas para descrever uma peça. Cada *feature* possui informações relativas à funcionalidade da sua forma geométrica relacionada ao projeto e manufatura;
- Uma forma geométrica ou entidade cuja presença ou dimensões são requeridas para a realização de pelo menos uma das atividades de um sistema CIM;
- Uma entidade capaz de armazenar informações do produto que podem ajudar a atividade de projeto assim como a comunicação do projeto com a fabricação ou entre quaisquer outras atividades de engenharia (Shah);
- Uma entidade manipulada durante atividades de projeto, engenharia e manufatura.

Em grande parte das definições apresentadas, busca-se estabelecer uma associação entre *feature* e forma geométrica. Isto se dá porque a maioria das aplicações que utilizam *features* é voltada para a área de planejamento de processo, onde a forma geométrica é essencial, e o termo *feature* é utilizado para se referir a *feature* de forma. Para que as *features* possam ser utilizadas como elemento de ligação entre as atividades do ciclo

produtivo devem ser capazes de armazenar informações geométricas que sejam necessárias para a realização de algumas etapas do ciclo produtivo.

2.1.2 Modelagem de peça baseada em *feature*

Neste tipo de abordagem, uma biblioteca de *features* de projeto é colocada a disposição do projetista, que cria a peça através da instânciação das *features* presentes nesta biblioteca (Shah, Bhatnagar e Hsião, 1988:489). Para representar o modelo de peça, nessa perspectiva, usa-se as *features* dessa biblioteca. No caso de se trabalhar com *features* em contextos diferentes, como os de projeto e manufatura, há a necessidade da conversão das *features* entre os domínios em tela para a modelagem de peça como por exemplo, na conversão das *features* do domínio de projeto para o de manufatura.

De um modo geral, a modelagem de peça baseada em *features* pode ser feito de diferentes maneiras:

- Um-para-um: quando a *feature* resultante da modelagem é idêntica a *feature* mapeada (do outro domínio);
- Re-parametrização variante: quando diferentes conjuntos de atributos são utilizados para representar a mesma *feature* em diferentes domínios;
- Agregação discreta: quando duas ou mais *features* de um domínio são mapeadas para uma única *feature* em outro domínio;
- Decomposição discreta: quando uma *feature* é modelada para duas ou mais *features* em outro domínio;
- Conjugação: quando uma *feature* (obtida após o mapeamento) é resultante de apenas algumas partes de duas ou mais *features* de um outro domínio.

Para se obter o modelo de peça para projeto, a partir da modelagem por *feature*, é necessário que se realize um projeto por *features*. A mediação das ferramentas informáticas em atividades de projeto como esse, pode se dar em três etapas:

- 1) Projeto conceitual ou preliminar;
- 2) Projeto estrutural ou de configuração;
- 3) Projeto paramétrico ou detalhado.

Sistemas como estes têm sido construídos, como os relatados em Shah, Hsião e Robinson (1990) e You, Chu e Kashyap (1989), mas são adequados somente para a etapa

de detalhamento. As etapas de projeto estrutural e conceitual ainda não dispõem de sistemas baseados em *features*.

Por outro lado, ao utilizar um sistema de projeto através da modelagem de peça baseado numa biblioteca de *features* de projeto, o projetista deve se limitar à utilização das *features* presentes nessa biblioteca. Entretanto, essa observância constitui-se numa desvantagem do projeto por *features* em relação ao reconhecimento de *features*.

2.1.3 Pesquisas sobre *features*

A tecnologia baseada nas tecnologias *Design by feature* está ainda no seu começo, sendo necessário realizar pesquisas voltadas a aplicações nas etapas do ciclo produtivo. No entanto, as pesquisas sobre essas tecnologias apresentadas aqui, convergem para os seguintes caminhos nas áreas que se seguem:

- Representação de *feature*: procura definir como as *features* serão representadas internamente no computador. Nessa direção, apontam-se dois aspectos relevantes:

- ☛ Forma: pode ser volumétrica ou superficial;

- ☛ Significado em engenharia: é uma área onde os métodos estão muito pouco desenvolvidos. Existem perspectivas relacionadas às formas geométricas para que as *features* sejam utilizadas para capturar significado em engenharia, durante todo o ciclo de vida do produto;

- Definição de *feature*: procura definir quais os atributos tanto geométricos quanto tecnológicos que devem estar associados às mesmas;

- *Features* e restrições: procura definir quais as relações entre as restrições que são impostas sobre a peça e as *features* que a constituem;

- Validação de *features*: procura determinar quais são as condições que devem ser satisfeitas para que as *features* sejam validadas. Por exemplo, um furo (*feature* negativa) não pode existir sem que esteja mergulhado num bloco ou eixo (*feature* positiva);

- Múltiplas visões sobre *feature*: estuda os mecanismos de conversão das *features* para diferentes domínios de aplicação. Esses mecanismos emergem das mais variadas visões de uso de peças, componentes ou sistemas mecânicos necessários para migração em aplicações com domínios desiguais. Por exemplo, pode ser necessária a análise de uma

mesma peça por software de CAE e CAPP. Certamente as informações necessárias a cada software seriam diferentes;

➤ **Padronização de *features*:** pesquisas nesta área buscam classificar e padronizar *features*. Os primeiros esforços neste sentido foram feitos para a padronização de *features* de forma e são relatados em CAM-I (1986).

Nesse trabalho, usa-se a proposta conceitual dos sistemas baseados nas tecnologias *Design by feature* aplicados entre as fases de projeto conceitual e preliminar da estrutura geral de projeto. Então, nessa perspectiva, uma *feature* é um conjunto de informações usadas para descrever as funcionalidades geométricas de uma peça a partir das suas formas topológicas – volumétricas, espaciais ou até mesmo planar.

2.2 Árvore de Estrutura hierárquica de funções de peça

Ao final das fases do projeto informacional e conceitual do produto, nas quais ocorre a determinação da função principal e conceitual do problema de projeto, faz-se necessário estabelecer a estrutura funcional e conceitual de cada uma das peças componentes desse problema.

No projeto conceitual da peça, tem-se a possibilidade de se disponibilizar as informações funcionais e conceituais das estruturas de dados hierarquizados através dos atributos e informações. Back & Forcelline, afirmam que esta etapa é a mais importante do ciclo de vida da peça, além das outras etapas de projeto como a manufatura, montagem e marketing. Além disso, atribui-se a essa etapa a responsabilidade de uma grande porcentagem do custo final do produto. Assim sendo, para se obter em tempo hábil a definição de uma peça, basta sistematizar as informações sobre a peça desejada. Espera-se nessa etapa do projeto reduzir custos, tempo e aumentar a qualidade do produto.

Dessa forma, de um modo geral, na árvore de dados hierarquizados de cada peça, suas restrições de funcionalidade e de interface com outras peças devem ser consideradas no conjunto das peças. As informações de restrições funcionais e de *layout* são de grande importância, a ponto de interferirem sobremaneira na forma geométrica de cada peça e assim influenciarem na própria estrutura conceitual, preliminar bem como, também, no próprio produto. Essas informações devem ser processadas de forma sistematizada, devendo compor o modelo de representação desejado (intencionado), sendo determinante no processo de projeto das peças, componentes e sistemas mecânicos de um produto.

2.2.1 Estrutural funcional de peça

A partir da proposta de 'Modelamento de dados para o desenvolvimento e representação de peças' de Linhares (2000:36), pode-se modelar uma peça numa estrutura funcional da maneira como mostrada na Figura 05.

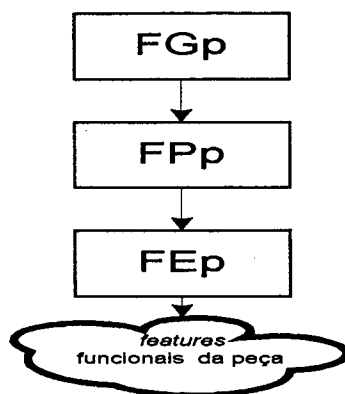


Figura 05: Estrutura funcional de peça (apud Linhares)

Conforme a Figura 05, os níveis hierárquicos da estruturação funcional tomam os significados mostrados a seguir.

Nível 2: função global de peça (FGp);

É a função principal de peça. É resultante da combinação de particularidades ou configurações geométricas diferenciadas no volume de material da peça. De certa forma, seria como se a peça fosse uma caixa preta definidora do 'como' as entradas são transformadas na saída desejada, obedecendo às interfaces de entrada e saída, nessa ordem.

Nível 1: função parcial de peça (FPp);

São os componentes formadores e principais da peça situados nas regiões principais da peça e cujos grupos funcionais são executados a partir dos princípios de soluções de peça.

As funções parciais são semânticas que buscam definir uma necessidade específica que a peça deve cumprir, podendo ser formada por uma ou mais funções elementares de peça.

Nível 0: função elementar de peça (FEp).

São funções mais básicas, realizadas por soluções construtivas que compõem os grupos funcionais da peça. São suportadas por *features* funcionais¹ de peça. Essas estão em um nível ainda mais primitivo.

O nível mais primitivo, o das formas geométricas, não será tratado aqui nesse trabalho pois, isso se constituirá ainda num trabalho de pesquisa futuro em nível de um sistema CAD.

2.2.2 Estrutura conceitual de peça

Segundo Linhares, a estrutura conceitual de peça é formada por uma região funcional e pelos requisitos de projeto da mesma. Na região funcional de peça (RFp) define-se a posição relativa no espaço disponível do volume do material. Nela deve ser realizada uma determinada função parcial de peça, com a corroboração dos seus grupos funcionais, nesse espaço topológico (partes do corpo da peça).

Os efeitos decorrentes do posicionamento de uma RFp são refletidos posteriormente no desempenho da FGp.

Os requisitos de projeto de peça (RPp) são atributos funcionais², geométricos³ e tecnológicos⁴ (como os atributos de feature) necessários ao cumprimento de sua FGp. Dependem dos parâmetros adotados para os atributos funcionais e conceituais do produto. Resumindo, os RPp são uma lista de considerações que identifica algumas particularidades acerca da peça em questão, bem como, reflete os próprios requisitos de projeto do produto sobre a mesma durante todo o ciclo de vida da peça.

Os princípios de solução de peça (PSP) são regras de soluções construtivas que permitem a configuração das FPp consideradas nos grupos funcionais da mesma. Na sua escolha deve-se levar em conta os critérios determinados pelos RPp e por restrições das informações não-geométricas (custos, fabricação, montagem, operação, manutenção e descarte).

¹ *Features* funcionais são semânticas que descrevem como a FEp pode ser realizada. Elas podem ser representadas por pelo menos uma FEp ou ainda uma FEp pode ser definida por uma ou mais *features* funcionais de peça.

² Atributos funcionais são semânticas traduzidas por uma sintaxe.

³ Atributos geométricos traduzem possíveis formas geométricas para a realização das necessidades funcionais da peça

⁴ Atributos tecnológicos são especificações de detalhes construtivos da peça necessários a sua conformação para realização da FGp.

Cada grupo funcional de peça (GFp) possui uma única FPp, composta por FEp mais básicas. As funções elementares são conformadas por *features* conceituais⁵ de peça, que no projeto preliminar de peça serão refinadas e receberam atributos geométricos e tecnológicos ou preliminares e detalhados a posteriori. Importa-se notar, que numa mesma peça pode existir vários GFp idênticos.

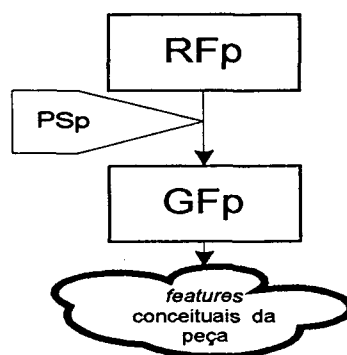


Figura 06: Estrutura conceitual de peça (apud Linhares)

De acordo com Linhares, a estrutura hierárquica da peça pode ser concebida da seguinte forma:

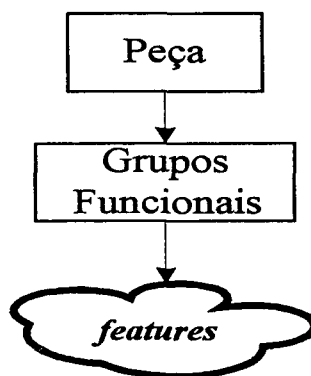


Figura 07: Estrutura hierárquica de peça (apud Linhares)

2.2.3 Exemplo de aplicação da árvore hierárquica de funções de peça

Na Figura 08, são mostrados a estrutura hierárquica de funções de peça com os seus correspondentes níveis funcionais, do maior para o menor e as categorias funcionais possíveis para implementação.

⁵ *Features* conceituais são designadas pelos PSp, já que estes irão configurar soluções possíveis para os GFp.

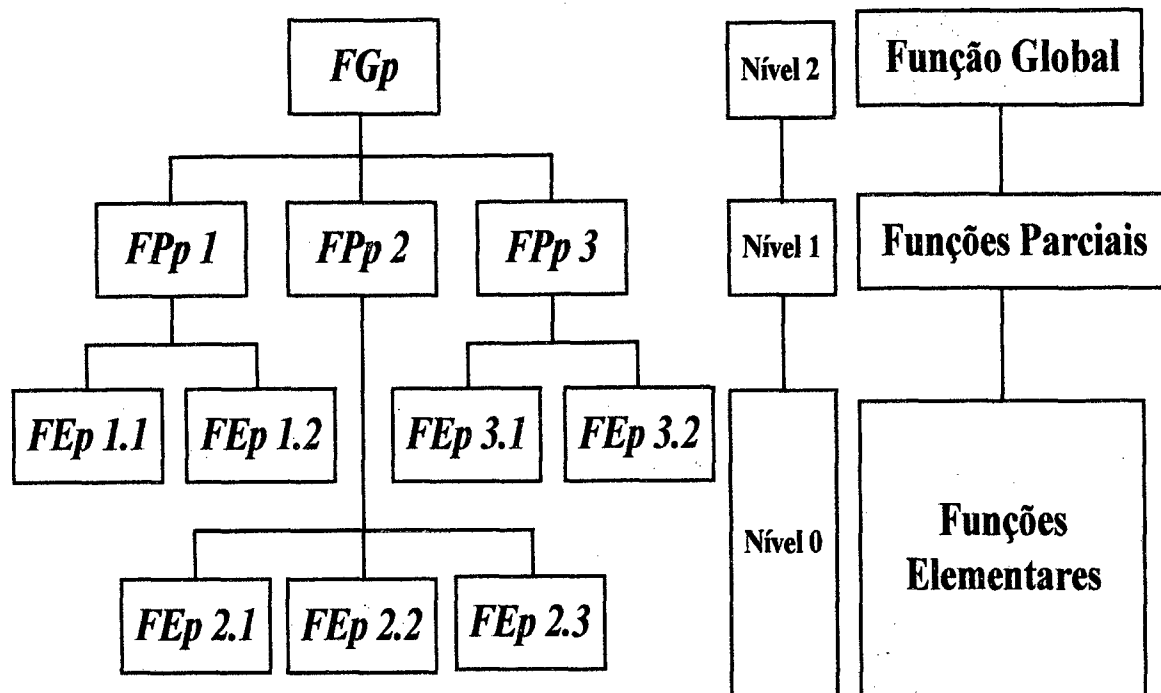


Figura 08: Correlação entre a estrutura hierárquica de funções, ordem dos níveis funcionais e categorias das funcionalidades de peça

Suponha que o projetista necessite de uma peça na elaboração de seu projeto e para isso ele tenta *featurizar* uma intenção morfológica da seguinte forma:

1º) Com muita objetividade: **“Preciso de biela.”**

E o sistema numa simples pesquisa lhe trouxesse uma lista de códigos de uma série de modelos de bielas (uma imensa família de bielas), isso não seria o máximo?

2º) Através das funcionalidades da peça:

FGp ← **“Transformar movimento orbital em movimento alternativo”**

FPp ← **“Acoplar radialmente a bucha de ligação”**

FPp ← **“Interligar anéis de acoplamento com haste tipo tronco de pirâmide”**

FPp ← **“Acoplar pino cilíndrico do pistão”**

Veja Figura 09

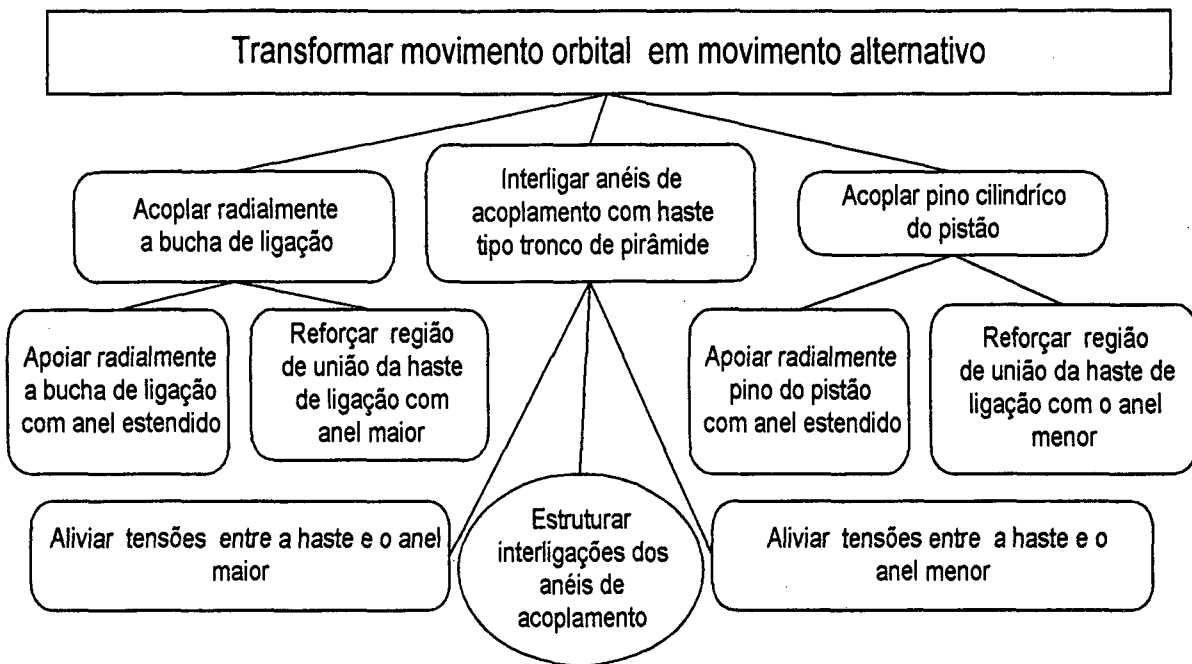


Figura 09: Desdobramento da função global de peça 'Transformar movimento orbital em movimento alternativo'

Suponhamos que cada funcionalidade correspondesse a uma *feature* específica e que cada *feature* fosse determinada por uma semântica a partir de um código. Então, o sistema poderia pesquisar e determinar uma lista de códigos ou um código em especial que agregasse todas as especificações solicitadas. Se não, ele poderia incluir nessa lista um outro conjunto de códigos cujas similaridades atendessem se não a todas as especificações, mas pelo menos uma grande parte das solicitadas. Isso também não seria o máximo?

Na próxima Figura, a de nº 10, tem-se uma árvore onde a raiz representa uma função global e os ramos, suas funções parciais e elementares. Cada nodo tem sua representação geométrica, cujo código expressa uma semântica específica, necessária e própria para o processo de projeto de peças, componentes ou sistemas mecânicos.

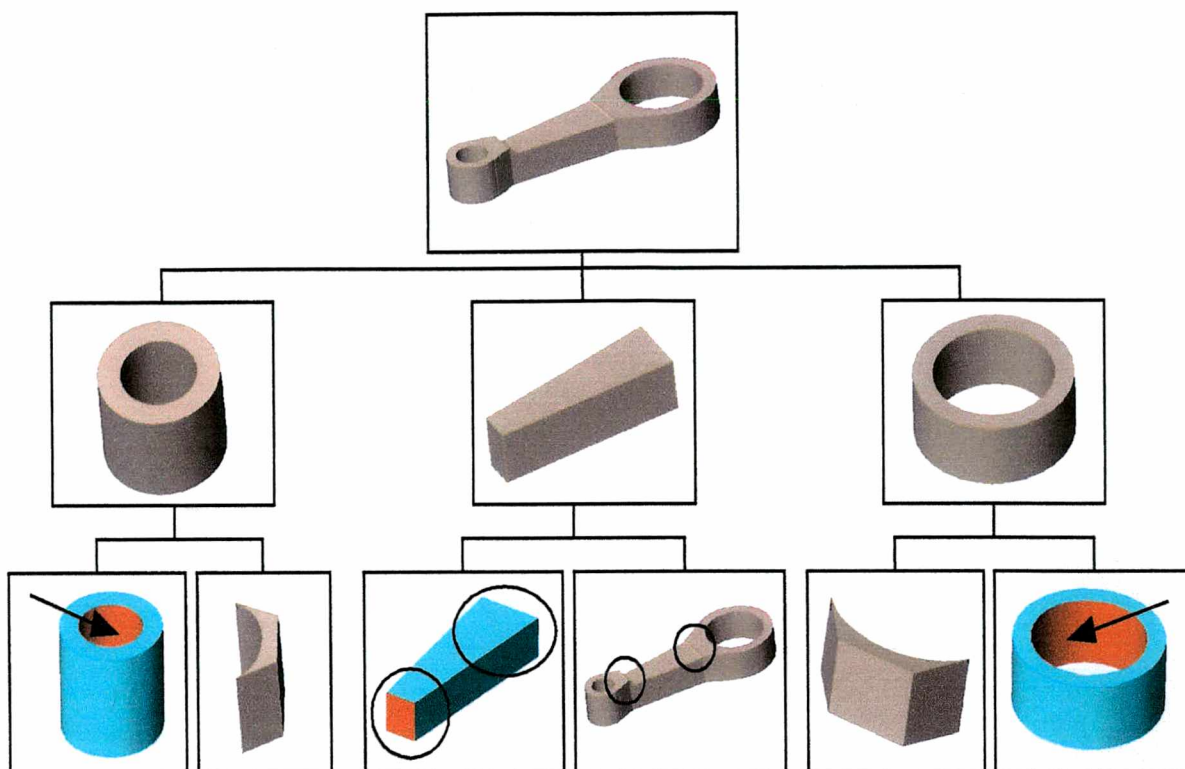


Figura 10: Árvore hierárquica com suas respectivas *featurizações*

2.3 Estado da arte

2.3.1 Processamento da Linguagem Natural - PLN

Para que o processamento da linguagem escrita, ou Processamento da Linguagem Natural (PLN) possa interpretar uma comunicação, é necessário, segundo Ribeiro (1987: 6) *“possuir o ‘conhecimento’ da estrutura das sentenças, do significado das palavras bem como de sua morfologia e das regras de conversação”*.

Entretanto, o problema básico do PLN é a transformação ou processamento da linguagem natural numa linguagem digital. Essa dificuldade gera sérios obstáculos em relação ao estabelecimento da comunicação à medida em que a Linguagem Digital (LD) não tem as características de sintaxe, semântica e pragmática da Linguagem Natural (LN).

Na LD a sintaxe se mantém, já a semântica só se torna formalizada através do uso adequado de Banco de Dados enquanto que a pragmática é totalmente inexistente. Isso faz com que qualquer transformação da LN - que é linear, discreta e polissêmica - para LD -

que é linear, discreta e monossêmica – acarrete em prejuízo de compreensão na transformação.

O PLN é uma área multidisciplinar envolvendo estudos nas áreas de ciências da computação, lingüística e ciências cognitivas. Ele é um ramo da Inteligência Artificial que tem por objetivo, através do uso de computadores, gerar e/ou interpretar textos em uma língua natural (p. ex: Inglês, Francês, Português, Espanhol, Italiano, etc).

Então, criar sistemas computacionais e processadores de linguagem natural é também uma forma de emergir ao mundo do falante da linguagem escrita e uma tentativa não só de compreender tal linguagem, mas de explorar o conhecimento que esse mundo pode nos proporcionar e dessa forma sustentar ou não nossas teorias, aprioristas que sejam.

As pesquisas em PLN dividem-se em duas áreas de trabalho: interpretação e geração:

a) interpretação de linguagem natural: baseia-se em mecanismos que tentam compreender frases em alguma língua natural, buscando mapear a informação recebida para uma representação mais diretamente utilizável pelo computador. Exemplos clássicos dessa interpretação são as interfaces em LN para consultas em Banco de Dados.

b) geração de linguagem natural: é o oposto da interpretação. O computador traduz uma representação interna para um texto em alguma língua. O texto de saída deve ser o mais próximo de um texto produzido por um falante daquela língua.

Todavia, faz-se necessário ter em mente que as dificuldades encontradas normalmente, são geradas pelas próprias propriedades que tornam a linguagem tão poderosa. Na Tabela 01, Rich & Knight (1993: 435) ressaltam aspectos positivos do processamento da linguagem que a tornam um fenômeno tão significativo.

PROBLEMA DA LINGUAGEM	ASPECTO POSITIVO
1) As frases de uma língua são descrições incompletas das informações que pretendem transmitir;	1. A linguagem permite aos que falam ser tão vagos ou precisos quanto quiserem. Ela também permite que eles deixem de mencionar coisas que acham que os ouvintes já sabem;
2) A mesma expressão significa coisas diferentes em contextos diferentes;	2. A linguagem pode evoluir juntamente com a evolução das experiências que queremos comunicar;
3) Nenhum programa de linguagem natural pode ser completo porque novas palavras, expressões e significados podem ser gerados com bastante liberdade;	3. A linguagem pode evoluir juntamente com a evolução das experiências que queremos comunicar;
4) Há inúmeras maneiras de dizer a mesma coisa.	4. Quando você sabe muito, os fatos simplesmente estão implícitos uns nos outros. A linguagem é para ser usada por agentes que sabem muito.

Tabela 01: Características da linguagem que a tornam tão difícil quanto útil, compiladas de Rich & Knight.

Apesar dos aspectos positivos acima apresentados como vantagens, existem outros problemas (desvantagens) relacionados à informação textual. Dentre estes problemas segundo ainda Rich & Knight, os mais difíceis de serem tratados são a ambigüidade e a falta de uma estrutura comum.

Mesmo assim, o objetivo das pesquisas sobre interpretação e geração de linguagens é criar modelos computacionais que sirvam como base para os programas de computadores cuja meta maior é realizar diversas tarefas envolvendo as línguas naturais e, em particular, as linguagens escritas.

Para cumprir o objetivo acima, ainda hoje, os trabalhos de análise de linguagem natural fazem uso de um modelo clássico sequencial, herdado da teoria lingüística, de fácil uso e manutenção, mas possuindo sensíveis limitações quanto ao seu desempenho. O modelo sequencial separa a análise em diferentes passos, cada um correspondendo determinado modelo lingüístico:

✓ **Processamento Morfológico**

De um modo geral, há dois tipos de análise: a formação da palavra e a da sentença. Na primeira, emprega-se o processo de decomposição da palavra em suas unidades de significação, ou seja, em seus elementos mórficos ou morfemas, tais como: radicais, tema, vogal temática, afixos (prefixos e sufixos) e desinências. Na segunda, o modo com o qual as palavras são construídas em termos de seus componentes, atribuindo traços sintáticos a cada uma das palavras de uma frase, tais como substantivo, adjetivo, advérbio, verbo, conjunção, preposição; gênero (masculino ou feminino) e número (singular ou plural), entre outros.

✓ **Processamento Sintático**

Etapa em que seqüências lineares de palavras são transformadas em estruturas que mostram como as palavras estão relacionadas entre si. Algumas seqüências de palavras podem ser rejeitadas se violarem as regras da linguagem sobre como as palavras podem ser combinadas.

✓ **Processamento Semântico**

Analisa o relacionamento das palavras e como elas se combinam para formar o significado da frase.

✓ **Processamento do Discurso**

Analisa o significado das palavras e como elas se combinam para formar o significado da frase.

✓ **Processamento Pragmático:**

Responsável em assegurar que o significado da frase seja consistente com os objetivos da pessoa que fala.

Todas as etapas descritas acima são importantes em um sistema completo de compressão da linguagem natural. No entanto, nem todos os sistemas são desenvolvidos exatamente com todas essas etapas. Às vezes, devido à complexidade da língua e a impossibilidade de se determinar regras de reescrita bem definidas, podem ser omitidas duas ou mais regras, resultando em um sistema bem mais simples de se implementar.

Em geral, as tarefas do PLN seja em interpretar ou gerar linguagem fonte, e até mesmo na tradução de uma língua para outra, são baseadas em processamento de texto

escrito e/ou processamento de linguagem falada envolvendo a comunicação homem-máquina. Alguns exemplos de aplicações são:

- ✓ Análise de documentos;
- ✓ Geração automática de textos;
- ✓ Interface para bases de conhecimento (bando de dados, sistema especialista);
- ✓ Revisão de textos no sentido de identificar e corrigir erros ortográficos e de concordância;
- ✓ Sistemas de atendimento automático a clientes através de telefone;
- ✓ Tradução automática de documentos de uma linguagem para outra;
- ✓ Sumarização de textos;
- ✓ Extração de informações de mensagens ou artigos em determinados assuntos;
- ✓ Sistemas para encontrar documentos apropriados sobre um determinado tópico a partir de um banco de dados de textos;
- ✓ Sistemas tutoriais, onde a máquina interage com o usuário;
- ✓ Sistemas para resolver problemas cooperativos (ex.: um sistema de ajuda de planejamento e calculo de cargas marítimas).

2.3.1.1 Alguns conceitos em lingüística

No dizer de Araribóia (1988:149) a “linguagem é todo sistema do qual é possível extrair uma significação, podendo servir como meio de comunicação”. Assim sendo, toda linguagem é composta de elementos ou unidades que se combinam de uma determinada forma para produzir uma significação.

Portanto, uma língua natural é formada por um conjunto infinito de orações (conjunto de frases). Uma frase pode ser dividida em unidades básicas, as palavras, de sons e significados. Essas, por sua vez, podem ser subdivididas em unidades mínimas de sons (fonemas) e de significados (morfemas).

O morfema básico de uma palavra é chamado de raiz. A uma raiz podem-se afixar outros morfemas, por exemplo, prefixos, sufixos, vogal temática e desinência.

A categoria gramatical é também conhecida como classe de palavras. As palavras são classificadas em categorias sintáticas de acordo com sua denotação. Uma mesma palavra pode pertencer a mais de uma categoria sintática, dependendo de sua posição dentro de uma oração. As classes mais conhecidas são: substantivo, adjetivo, numeral, pronome, verbo, advérbio e conectivo.

a) Frase como estrutura

Toda frase de uma língua consiste em uma organização, uma combinação de elementos lingüísticos agrupados segundo certos princípios, que a caracterizam como uma estrutura.

A organização interna, implícita na estrutura frasal, obedece a princípios gerais bem definidos, a partir dos quais o falante é capaz de dizer se uma seqüência de palavras: a) está de acordo com o sistema gramatical da língua, isto é, se essa seqüência pode ser obtida através da aplicação das regras da gramática; b) se ela se apresenta completa ou incompleta; c) se é passível de interpretação semântica, etc. Pode-se evidenciar, ainda, que tais frases se constituem de classes de elementos equivalentes quanto às funções que nelas vêm a desempenhar.

Ao se observar textos, de um modo geral, é comum se permitir chegar, intuitivamente, a algumas conclusões elementares, com as apresentadas a seguir:

- toda leitura de texto, evidentemente, se inicia no princípio de uma frase e se conclui no fim de outra frase;
- a escolha do texto obedece a uma intuição automática, isto é, sempre se seleciona uma seqüência verbal cujo início coincide com o começo de uma frase e cujo fim coincide também com o término de uma outra frase;
- se casualmente se interromper essa mesma seqüência por qualquer razão que seja, a intuição humana imediatamente chamará a atenção para a ocorrência de alguma anormalidade.

Isso porque todo texto é um conjunto (por vezes, unitário) de frases, que constituem uma unicidade semântica/pragmática: semântica, por introduzir significados distintos daquele de cada uma das frases tomada isoladamente e pragmática, por constituir um ato de comunicação, destinado a atuar sobre o ouvinte/leitor de determinado modo.

Ora, no ambiente de processo de projeto de peças, componentes ou sistemas mecânicos, assim como no dia-a-dia, toda frase diz algo, fala sobre um determinado estado de coisas do mundo em tela, mas o faz de uma maneira singular e contextualizada pelo falante/comunicador.

Assim, os constituintes da estrutura da frase, o conteúdo proposicional, são propalados por meio de elementos lingüísticos – fonemas, morfemas, vocábulos – selecionados dentre os inventários que cada língua oferece (paradigma) e combinados de acordo com certos princípios de organização (sintagmas). Para Souza e Silva, M. C. P. & Koch, *“o material lingüístico assim estruturado recebe o nome de proposição (P) ou oração (O) e pode ser veiculado pelo locutor sob o modo da asserção, da pergunta, da ordem, etc. Aos diversos ‘modos de dizer’ é que se denomina tipo de frases (T). Logo, a primeira regra de constituição de toda e qualquer frase (F) de uma língua pode ser determina por: $F \leftarrow T + P$ ”* (2000,12).

Desse modo, todas as línguas podem ser descritas a partir de um conjunto de fórmulas semelhantes (denominadas regras de reescritas, de estrutura frasal ou de base) cuja função consiste em indicar a estrutura subjacente, elementar e abstrata dos elementos que compõem a proposição, especificando e formalizando as relações de dominância e de procedência existentes entre esses elementos.

b) Definição de frase

Frase: *“construção que encerra um sentido completo, podendo ser formada por uma ou mais palavras, com verbo ou sem ele, ou por uma ou mais orações; pode ser afirmativa, negativa, interrogativa, exclamativa ou imperativa, o que, na fala, é expresso por entonação típica e, na escrita, pelos sinais de pontuação”*. (Houaiss: 2001,1387)

Função Sintática: Sujeito, objeto, complemento, adjunto, conectivo, etc;

Papéis Temáticos: Papéis semânticos comuns em formas sintáticas alternativas: agente, paciente, instrumentos, etc.

c) Princípios de organização da estrutura frasal

Esses princípios têm como objetivo (tentam) descrever a estrutura da frase via as regras de reescritas, de estrutura frasal ou de base através da determinação - em unidades menores e, também, da verificação de equivalência entre elas - dos elementos aptos a compor a proposição das frases do português, observando a sua organização em classes e as suas possibilidades combinatórias.

Para tanto, segundo Souza e Silva, M. C. P. & Koch I. V. e Fortkamp, M^a B. M. & Tomitch, L. M^a B., utiliza-se o procedimento da comutação, cujas tarefas básicas são:

Procedimento da comutação	Objetivo
1. segmentação	determinar os subconjuntos em que pode ser decomposta a proposição
2. substituição	verificar quais desses subconjuntos exercem a mesma função

d) Diferentes processos de segmentação

Segundo as pesquisas de Fortkamp, M^a B. M. & Tomitch, L. M^a B (2000:204-207), “A segmentação de uma frase em partes menores pode ocorrer em vários níveis de abrangência”.

Os autores ainda destacam três níveis de abrangência: i) nível mais alto, segmentado pela oração, envolvendo um sintagma verbal e seus argumentos (sujeito, objeto, etc); ii) nível intermediário, segmentado por diferentes sintagmas e palavras e iii) nível mais baixo, segmento envolvendo fonemas.

Entretanto, os mesmos autores, sugerem três técnicas comumente usadas na segmentação de frases complexas (Cf. Fortkamp, M^a B. M. & Tomitch, L. M^a B, 2000:207-210):

1^a) iniciar pela primeira palavra da frase e avançar da esquerda para a direita palavra por palavra.

Essa técnica é essencialmente um modelo de processamento *bottom-up*, linear por processar palavra por palavra. Por ser gerado através de uma gramática de estados finitos é

incapaz de resolver o problema de orações intercaladas, devido sua incapacidade de gerar todas as frases possíveis da língua.

2ª) iniciar pelo marcador formal, geralmente uma conjunção, que identifica o início da oração, coordenada ou subordinada e, a partir daí, busca, à direita, o fim da oração;

Essa outra técnica de segmentação pressupõe um processamento *top-down*. Conta com um catálogo de modelo que pode ser seguido, principalmente, para as orações coordenadas e subordinadas além de mostrar melhor desempenho em frases complexas quando as orações que eles marcam não estão invertidas e nem intercaladas dentro de outras orações (Cf. Fortkamp, Mª B. M. & Tomitch, L. Mª B, 2000:209).

3ª) iniciar pelo sintagma verbal, identificando o tipo de verbo e depois localizar o sujeito e os componentes, quando exigidos pelo verbo.

Nessa última técnica, segundo os mesmos autores *“o verbo não é apenas a razão de ser da oração mas o centro de onde emanam os controles a cada um de seus argumentos, cujo número varia de acordo com a valência de cada verbo, podendo variar de zero a quatro”*. Entretanto, devido à ambigüidade valencial de certos verbos como chover e amar, que ora funcionam como transitivo direto ora como indireto, possibilita diversas segmentações (Cf. Fortkamp, Mª B. M. & Tomitch, L. Mª B, 2000:210).

e) Sintagmas como constituintes operacionais

O termo sintagma significa pura e simplesmente agrupamento.

Os sintagmas organizam-se em torno de um elemento fundamental, denominado núcleo, que pode, por si só, constituir o sintagma. Portanto, a natureza do sintagma depende do tipo de elemento que constitui o seu núcleo.

f) Conceito e classificação de sintagma

Sintagma é uma unidade da análise sintática composta de um núcleo (p. ex., um verbo, um nome, um adjetivo etc) e de outros termos que a ele se unem, formando uma locução que entrará na formação da oração [O nome do sintagma depende da classe da palavra que forma seu núcleo, havendo assim sintagma nominal (núcleo substantivo), sintagma verbal (núcleo verbo), sintagma adjetival (núcleo adjetivo), sintagma

preposicional (núcleo preposição); na teoria gerativa existem sintagmas formados por núcleos mais abstratos, como tempo, concordância etc] (Houaiss: 2001,2581).

Assim, revisitando a classificação dos sintagmas, tem-se:

- a) Os sintagmas nominais (SN) têm como núcleo um substantivo;
- b) Os sintagmas verbais (SV) possuem um verbo como núcleo;
- c) Os sintagmas preposicionais (SP) começam com uma preposição, etc;
- d) Os sintagmas adjetivais (SA⁶) têm como núcleo um adjetivo que, à semelhança do que ocorre nos demais tipo de sintagma, pode vir sozinho ou acompanhado de outros elementos; *intensificadores*⁷ e modificadores adverbiais (SP_A), antepostos ao núcleo, e sintagmas preposicionados (SP_C), pospostos a ele.

Exemplos:

SN: Eixo excêntrico, Eixo de transmissão, Eixo de transmissão excêntrica;

SV: Transmitir torque e rotação, Succionador de óleo de lubrificação, Propulsor do eixo excêntrico;

SP: para amortecimento do bloco (pré-especificada), da borracha do amortecedor, de apoio e amortecimento do bloco (pré-especificada);

SA: Interligar anéis '*de acoplamento*' com perfil regular (Adjetivo - SA);

SA: Interligar '*fortemente*' os anéis '*de acoplamento*' com haste perfilada (Intensidade Adjetivo - SA);

SA: Interligar '*temporariamente*' anéis '*de acoplamento*' com haste de ligação (SP (tempo) – Adjetivo // SA);

SA: Acoplar bucha de ligação '*radialmente forte*' [SP (modo) – Adjetivo // AS];

SA: Acoplar pino cilíndrico do pistão '*radialmente forte*' à base do módulo realizável. (Intensidade + Adjetivo + SP_C - SA);

⁶ Cf Souza & Silva: 2000, p. 22, com exemplos adaptados.

⁷ *Intensificadores* ou *advérbios de intensidade* são expressões que têm por função intensificar a idéia expressa pelo verbo, pelo adjetivo ou por outro advérbio.

2.3.2 Conhecimento e Linguagem

O processo de compreensão da linguagem natural deve ser capaz de raciocinar com o conhecimento contido nas expressões lingüísticas. Ele também deve explorar a estrutura da língua como representação-objeto, através da combinação das palavras e de seus significados na sentença, dentre outros aspectos. No entanto, um programa não pode simular o comportamento lingüístico sem primeiro levar em conta importantes aspectos da inteligência humana – como o conhecimento do mundo e suas habilidades de raciocínio. Por exemplo, para responder perguntas ou participar de uma conversa, uma pessoa precisa conhecer a língua falada e o contexto da conversa em particular. Logo, um sistema de linguagem natural necessitaria de métodos de codificação para usar este conhecimento da língua de forma a produzir o comportamento apropriado. Além disso, o conhecimento da situação atual (ou contexto) representa um papel crucial na determinação do ‘como’ interpretar uma sentença particular.

Para Rich & Knight; Araribóia; Le; Coulon e Scaruffi, todos, de um modo geral, ainda seguem a teoria dos sinais (1955) do semiólogo Morris segunda a qual *“as diferentes formas de representar o conhecimento são tradicionalmente classificados em: análise morfológica, sintática, semântica, pragmática e integração do discurso”*.

Em Rodrigues Filho (2000:20), *“a representação do conhecimento é um conjunto de mecanismos usados para armazenar e manipular o conhecimento”*. Conhecimento esse, que normalmente se apresenta sob a forma de objetos, classes, categorias e descrições de objetos. (exemplo: Eixo excêntrico é uma peça mecânica, ele produz rotação), evento (ex: rotação e translação são movimentos), conhecimento sobre como fazer determinadas coisas (ex: transformar movimento rotativo em movimento alternativo).

Rodrigues Filho ainda aponta algumas formas de representação do conhecimento atualmente em uso como as regras de produção, frames, redes semânticas, lógica, listas, árvores, e outras, salientando que o uso de uma ou outra forma dependerá do assunto (escopo do problema), do problema a resolver e dos recursos disponíveis para tal.

Entretanto, as definições de cada categoria de conhecimento não têm se apresentado como precisas e completas. Por exemplo, qualquer fato particular pode incluir aspectos de diversos níveis diferentes.

2.3.3 Etapas do Processamento de Linguagem Natural

A análise de linguagem, em geral, é dividida em cinco etapas, descritas sucintamente a seguir.

2.3.3.1 Processamento Morfológico

O processamento morfológico (também denominado como análise léxica), analisa cada palavra isoladamente em relação aos seus componentes, atribuindo categorias sintáticas e outros atributos a cada uma das palavras de uma sentença, ou seja, a meta é isolar cada componente (string, i. é, uma sequência de caracteres) presente no texto fonte e classificá-lo segundo uma classe sintática (substantivo, adjetivo, verbo, advérbio, conjunção, preposição, artigo), a partir da sua categoria sintática (sujeito, objeto direto, complemento nominal, adjunto adverbial e assim por diante).

Ressalta-se que uma mesma palavra pode denotar entidades diferentes do mundo ou até mesmo dentro do seu contexto, tornando o processo de escolha do significado difícil de ser realizado contando apenas com a palavra isolada. A essa dificuldade dá-se o nome de **ambigüidade léxica** e ela pode ser atenuada através de um processo denominado de “desambiguação léxica”: associando-se a cada palavra do léxico⁸ informações sobre os contextos em que pode aparecer cada sentido da palavra ou até mesmo através do uso dos traços sintáticos expressos pelos atores no seu contexto.

Porém, o uso de informações ou marcadores sintáticos em alguns contextos carece de uma pesquisa de campo mais elaborada e minuciosa. A exemplo, tem-se a falta de padronização ou consenso do uso mais adequado dos simples constituintes como (no contexto de projeto de peças, componentes e sistemas mecânicos) ranhura, ranhurada, sulco, sulcado, fenda, fendada, rebaixo, rasgo, reentrância, cavidade, protusão, saliência e extremidade.

Qualquer referência ao uso ou relacionamento de algum dos constituintes citados anteriormente a algum dos aspectos construtivos ou constitutivos de peça mecânica causa uma verdadeira panacéia. E no ambiente de projeto de componentes, peças ou sistemas mecânicos, a denominação e uso de tais palavras, independentemente do domínio do

⁸ Conjunto de palavras reconhecidas pelo analisador léxico. Também chamado de “dicionário”.

projeto, ficam exclusivamente a cargo do especialista da área e da taxionomia do sistema de codificação & classificação de peça empregado.

2.3.3.2 Processamento Sintático

Sendo as sentenças de uma linguagem, seqüências de símbolos do seu alfabeto, então existem inúmeras combinações possíveis desses mesmos símbolos. Entretanto, certamente em todas as combinações (de símbolos da linguagem) há sentenças pertencentes à linguagem. Por exemplo, considerando o domínio de componentes, peças e sistemas mecânicos, nem todas as combinações dos seus lexemas⁹ são sentenças válidas nesse domínio, pois a linguagem usada nesse ambiente segue a um conjunto de regras, ou deveria seguir, da língua Portuguesa. Via de regra, uma frase em português começa pelo sujeito seguido do predicado.

A forma como os diferentes símbolos do alfabeto se combinam numa sentença constitui a sintaxe da linguagem. A sintaxe de uma linguagem pode ser especificada por um conjunto de regras, designadas regras sintáticas. Uma sentença que obedece às regras sintáticas de uma linguagem, diz-se sintaticamente válida ou bem formada.

O processamento sintático tem por objetivo criar uma descrição estrutural da frase com o intuito de garantir que a mesma esteja bem formada. Esse processo também denominado de análise (*Parsing*), é realizado através da exploração dos resultados do processamento morfológico, convertendo a lista de palavras que formam a sentença em uma estrutura que defina as unidades sintáticas representadas pela lista.

Embora haja muitas maneiras de produzir uma análise, quase todos os sistemas usam, de um modo geral, dois componentes principais: uma representação declarativa denominada de gramática e um procedimento chamado de analisador. A maneira mais comum de representar uma gramática é através de um conjunto de regras de produção. Essas regras são utilizadas pelo analisador para compará-las com as sentenças de entrada e produzir as estruturas de representação.

⁹ Lexema é uma unidade de base do léxico, que pode ser um morfema, palavra ou locução (Cf. Houaiss: 2001,1750).

Para cada sentença analisada, o analisador (*Parser*) gera uma árvore sintática (ou mais de uma se a sentença for ambígua) segundo uma gramática específica. Dessa forma, um analisador mapeia um conjunto de palavras, a lista de palavras, em um conjunto de padrões sintaticamente significante e de acordo com as regras de produção. Por exemplo, a sentença “*Preciso de biela de ligação para excêntrico*” gera a árvore sintática da Figura 11.

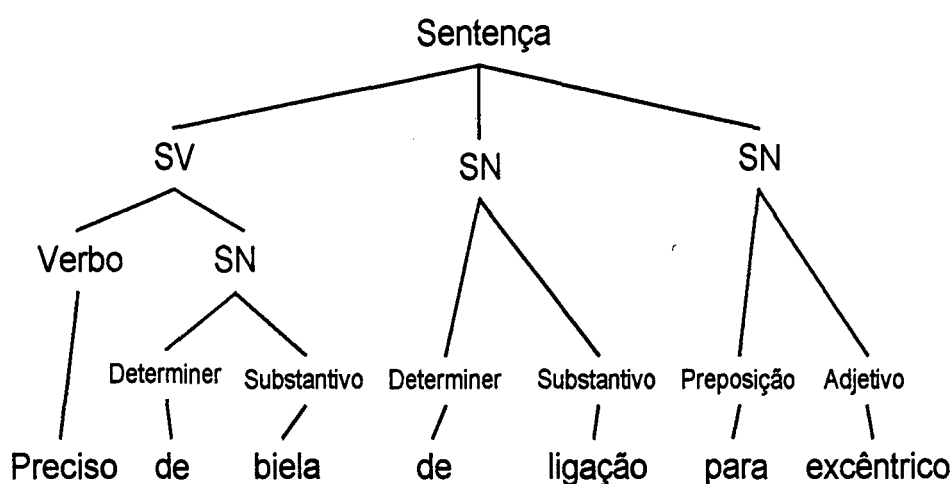


Fig. 11: Árvore Sintática: ‘Preciso de biela de ligação para excêntrico’

2.3.3.3 Processamento Semântico

Semântica¹⁰ é, em sua origem, o adjetivo correspondente a “sentido” - uma mutação semântica é uma mutação de sentido, o valor semântico de uma palavra é o seu sentido. O significado de uma sentença constitui a semântica da sentença. Nesse trabalho, a semântica é basicamente dependente do significado dado pelo projetista de componentes, peças e sistemas mecânicos a partir de sua necessidade, do domínio da peça que aqui se restringe ao domínio dos compressores alternativos e principalmente da taxonomia imposta pelo sistema de codificação & classificação KK-3.

A análise semântica faz a marcação dos significados das estruturas criadas pelo analisador sintático. Ela constitui-se como o próprio processo de mapeamento entre as estruturas sintáticas e os significados no domínio de atividade, rejeitando ou validando as construções derivadas de uma análise sintática preliminar.

¹⁰ Palavra formada do grego “sêmanô” (significar)

A análise semântica também pode ser considerada como o processo de associação de uma ou mais representações semânticas com árvore de derivação provenientes de uma análise sintática ou, até mesmo, de árvore em construção.

Diversas dificuldades estão associadas à representação do significado e do sentido. Segundo as pesquisas de Rich & Knight (1993: 459) *"várias abordagens foram desenvolvidas na busca de uma solução para o problema da criação de uma representação semântica para uma sentença"*. Dentre elas, destacam-se as seguintes:

➤ **Análise Conceitual:** a análise sintática é subordinada à análise semântica. Os conhecimentos sintáticos e semânticos são combinados em um sistema de interpretação único, o qual é guiado pelo conhecimento semântico.

➤ **Gramáticas de Casos:** a estrutura construída pelo processamento sintático contém algumas informações semânticas, apesar de ser necessária uma interpretação posterior.

➤ **Gramáticas Semânticas:** são gramáticas livres de contexto, onde a escolha de símbolos não terminais e de regras de produção são governadas por funções semânticas e sintáticas.

➤ **Interpretação Semântica Composicional:** o processo semântico é aplicado ao resultado de um processamento sintático. Isto pode ser feito de forma incremental, conforme a criação dos componentes, ou de uma só vez, quando a estrutura correspondente à sentença completa tiver sido construída.

➤ **Lógica de Predicados:** abordagem clássica para a representação de conhecimento desenvolvida por filósofos e matemáticos como um cálculo de processos de criação de inferências a partir de fatos.

➤ **Traços Semânticos:** consiste em associar propriedades semânticas a substantivos, adjetivos e verbos, indicando as categorias semânticas às quais a palavra está associada.

Não existe um consenso sobre qual o melhor método para determinado caso e, sim, um conjunto de propostas de solução.

2.3.3.4 Processamento do Discurso

Esta etapa do processamento estuda os aspectos necessários para capturar o significado de uma sentença. Dentre os muitos relacionamentos importantes que podem ocorrer entre sentenças e partes de contextos de seus discursos, pode-se citar a *elipse* e a *anáfora*.

A *elipse* é a omissão de palavras numa sentença, onde as mesmas aparecem subentendidas pelas sentenças emitidas antes. Por exemplo, “Preciso transformar movimento orbital, para comprimir gás refrigerante” (transformar movimento orbital em quê?).

A *anáfora* é a referência, usando pronomes, a uma entidade já antes definida. *Anáfora* entre sentenças é a referência pronominal a entidades já definidas em sentenças anteriores. Por exemplo, “Preciso transformar movimento alternativo em movimento orbital”. Ele acionará o pistão comprimindo o gás refrigerante.

2.3.3.5 Processamento Pragmático

O processamento pragmático é responsável em assegurar que o significado da frase seja consistente com os objetivos da pessoa que fala. Várias técnicas foram desenvolvidas no intuito de representar o conhecimento sobre como as pessoas agem e falam. Dentre elas, pode-se citar a modelagem de crenças e os postulados convencionais.

O processamento pragmático é uma espécie de tomada de decisões, considerando a interpretação obtida através da análise da sentença. Nesta etapa, decide-se o que fazer como resultado do processo de compressão. A figura 12 mostra um exemplo desse processamento, através da busca empreendida por um sistema de pesquisa em um banco de peças. Essa busca se dá, a partir da função global de peça “Necessito transmitir movimento orbital à biela” que deverá ser traduzida para a linguagem simbólica a qual fornecerá o código discriminador do tipo de peça em um domínio específico.

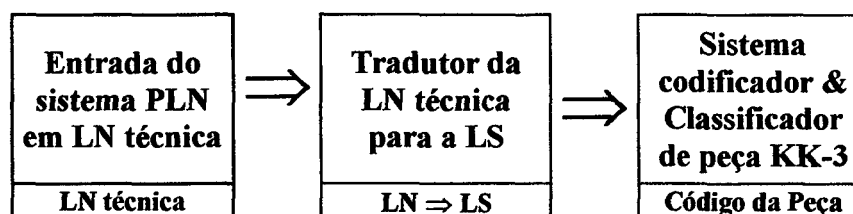


Figura 12: Esquema em bloco do Parser

2.3.4 Gramática e Parsing

Para Sethi (1995:77), “as gramáticas são capazes de descrever a maioria, mas não a totalidade das sintaxes de uma língua, pois uma parte limitada da análise sintática é realizada pelo analisador léxico”. Em programas de PLN a gramática é utilizada para ajudar a determinar os significados de uma sentença, e assim obter uma resposta apropriada. Vários tipos diferentes de gramáticas são usados em programas de LN, incluindo gramáticas de constituintes imediatos, gramáticas transformacionais, gramáticas de casos, gramáticas sistemáticas e gramáticas semânticas.

O *parsing* faz uso de regras gramaticais e outras fontes de conhecimento para determinar as funções das palavras na sentença de entrada (uma cadeia linear de palavras) a fim de criar uma estrutura de dados conhecida como árvore de derivação. Esta estrutura ilustra algumas das relações entre palavras nas sentenças e pode ser usada para obter o significado da sentença.

2.3.5 Gramática Formal

O conceito de gramática formal foi desenvolvido por Chomsky a partir de 1955. Esse desenvolvimento ocorreu na tentativa de aplicar os conhecimentos da lógica matemática e, em especial, as funções recursivas de Gödel, Church, Turing, Post, Kleene e outros, às línguas naturais para possibilitar a representação num corpo teórico e rigoroso, sistemático e integrado dos resultados da análise lingüística empírica.

Uma linguagem formal é um conjunto finito ou infinito de sentenças ou cadeias de caracteres, com comprimento finito, obtido por concatenação sobre um alfabeto.

Cada frase ou sequência de cadeias de caracteres desta língua é o produto das combinações dos elementos deste alfabeto entre si. Por exemplo, seja A um alfabeto e L uma linguagem desse alfabeto.

Então, se $A = \{a, b, c\}$

Logo, $L = \{a, b, aa, aba, ac, bac\}$.

Para gerar um número infinito de frases, necessário nas línguas naturais, introduz-se o mecanismo de recursividade. Recursividade pressupõe a definição de um conjunto finito de funções $f_1, f_2, f_3, \dots, f_n$. A estas funções são associadas regras de equivalência

entre diferentes partes das frases de uma língua, o que possibilita derivar um conjunto infinito.

Assim, por exemplo, consideradas as frases f_1 e f_2 abaixo:

(f_1) $abc \sim a$

(f_2) $ab \sim ba$

Define-se que o fragmento, a cadeia de caracteres ou o token, “abc” é equivalente ao fragmento “a” e que “ab” equivale a “ba”. Aplicando as regras recursivamente à frase “a” tem-se:

a

abc regra (i)

bac regra (ii)

babcc regra (i)

Dizer que uma língua é recursivamente enumerável, como o exemplo acima, é dizer que ela pode ser definida por algum tipo de sistema formal, tal qual a máquina de Turing, que é o sistema formal mais geral que se conhece, conforme a hipótese de Church. Desde então, a classe das linguagens enumeráveis recursivamente representa o conjunto de todas as linguagens que podem ser reconhecidas mecanicamente em um tempo finito.

Portanto, a sua definição matemática é uma quádrupla do tipo $G = (A_n, A_t, R, S)$, onde:

A_n é um conjunto finito de símbolos não terminais;

A_t é um conjunto finito de símbolos terminais;

R é o conjunto finito de regras de produção

S é símbolo inicial da gramática..

Possuindo as seguintes restrições:

- $A_n \cap A_t$ deve ser vazio
- $S \in A_t$
- O vocabulário (V) é composto por $A_n \cup A_t$

➤ O conjunto $(A_n \cup A_t)^*$ representa a linguagem definida pelo fechamento transitivo de Kleene da união das linguagens N e A_t mais o elemento vazio (ϵ)

➤ As produções são do tipo $P = \{ \alpha \rightarrow \beta \mid \alpha \in V^+ \text{ e } \beta \in V^* \}$, onde V^* é o vocabulário da linguagem incluindo a sentença vazia (ϵ) e $V^+ = V^* - (\epsilon)$

Portanto, a gramática pode ser exemplificada como na Figura 13.

<sentença>	→	<sintagma_nominal> <sintagma_verbal>< sintagma_nominal> <sintagma_preposicional>
<sintagma_nominal>	→	determiner, substantivo substantivo substantivo, adjetivo
<sintagma_verbal>	→	verbo
<sintagma_preposicional>	→	preposição ε

Fig. 13 - Uma gramática simples para uma linguagem natural

A Figura 13 mostra uma estrutura gramatical simples e livre de contexto. A barra vertical, “|”, deve ser lida como “ou”. O “ε” denota a cadeia de caracteres vazia. O símbolo inicial é <sentença>, gerador dos demais símbolos. Os símbolos não terminais, entre os sinais de menor que e maior que representam os símbolos que ainda podem ser expandidos pelas regras. Os símbolos terminais, em negrito, representam os símbolos que correspondem diretamente às cadeias de caracteres ou strings que têm de ser encontradas em uma frase de entrada.

Então, a linguagem gerada por uma gramática, $L(G)$, é o subconjunto de todas as sentenças que podem ser geradas a partir do símbolo inicial desta gramática, através de derivações sucessivas.

O formalismo gramatical pode ser classificado por sua capacidade gerativa, que é o conjunto de linguagem que uma gramática pode representar. Existem quatro tipos de gramática que diferem somente na forma das regras de reescrita. Uma síntese dos formalismos gramaticais, conhecida como hierarquia de Chomsky, é mostrada a seguir (confira na Figura 14):

➤ *Tipo 0* ou Gramática Irrestrita: Nenhuma restrição é imposta sobre a forma ' $\alpha \rightarrow \beta$ ', o que faz com que este tipo de gramática possa gerar qualquer linguagem. As linguagens geradas por este tipo de gramática são exatamente aquelas reconhecidas pela Máquina de Turing. Entretanto, apesar de computáveis são indecidíveis;

➤ *Tipo 1* ou Gramáticas Sensíveis ao Contexto (GSC): Tem as produções na forma ' $\alpha \rightarrow \beta$ ' onde ' $|\alpha| < |\beta|$ ', ou seja, o lado direito da produção deve conter pelo menos a mesma quantidade de símbolos presentes no lado esquerdo. São indecidíveis;

➤ *Tipo 2*: Chamada Gramática Livre de Contexto (GLC): As produções são do tipo $P = \{ \alpha \rightarrow \beta \mid \alpha \in A_n \text{ e } \beta \neq \epsilon \}$. O lado esquerdo da produção deve conter exatamente um símbolo, sendo este um não terminal. É a mais popular das gramáticas para as LN. Este tipo de gramática possui complexidade polinomial;

➤ *Tipo 3* ou Gramática Regular (GR): As produções são do tipo $P = \{ A \rightarrow aX \mid A \in A_n \text{ e } a \in A_t \text{ e } X \in (A_n \cup \{\epsilon\}) \}$. As linguagens geradas são exatamente aquelas reconhecidas por um autômato de estado finito. Este tipo de gramática possui complexidade linear.

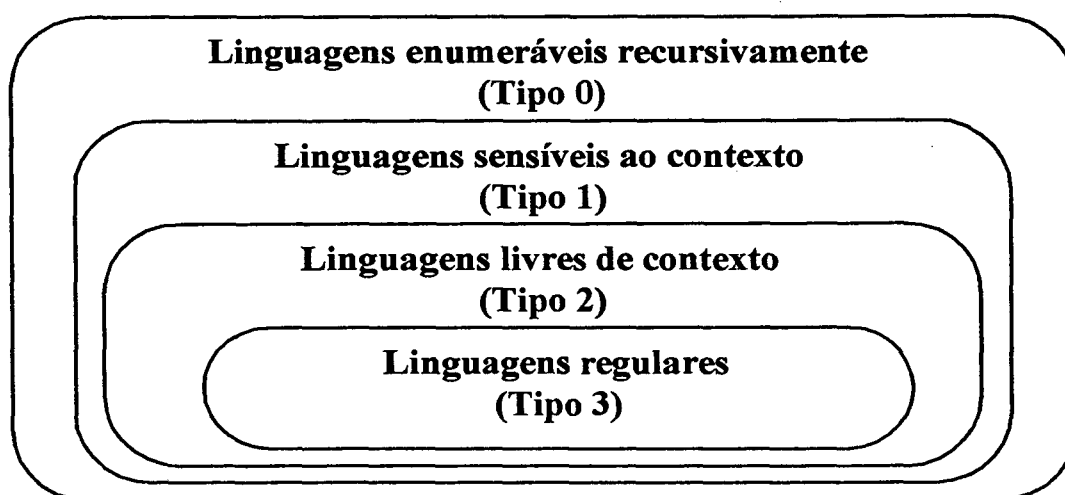


Figura 14: Hierarquia de Chomsky

3 MODELO DE *PARSER* PARA O PROCESSO DE PROJETO EM AMBIENTE DE SISTEMA MECÂNICO

Para os socráticos, tudo são números. Na programação em lógica, tudo são listas.

Ilson W. Rodrigues Filho

3.1 Hierarquia: definição, níveis e complexidade

Hierarquia, para Houaiss (2001:1531), é uma *“organização fundada sobre uma ordem de prioridade entre os elementos de um conjunto ou sobre relações de subordinação entre os membros de um grupo”*. Portanto, a organização reflete a descrição de uma estrutura de sistema e a ordem de prioridades ou relações de subordinação entre os seus elementos, os seus níveis.

Então, a hierarquia na análise de sistemas é uma categorização das características implícita nos elementos [caractere, cadeia de caracteres (string) ou lista] da estrutura da sentença.

Assim, como um conjunto é formado de elementos suscetíveis de possuírem certas propriedades em comum e de terem entre si, ou com outros elementos de outros conjuntos, certas relações, assim são os níveis de uma estrutura hierárquica. No nível dos caracteres, cadeias de caracteres ou listas (conjuntos de strings) os graus de complexidade da estrutura em cada um deles podem diferir-se, a ponto de termos relacionamentos de nível de menor grau de complexidade com outro de maior grau de complexidade ou vice-versa (Cf. Rodrigues Filho, 2000:38).

Os elementos constituintes dos níveis hierárquicos classificam-se como unidades simples ou compostas. Os primeiros são assim chamados por pertencerem ao mesmo nível hierárquico, por exemplo, o nível mais elementar, o dos caracteres, é formado pelas letras do alfabeto, os algarismos e os símbolos operadores e especiais; o das cadeias, pelas combinações dos elementos do nível dos caracteres e finalmente o nível das listas que podem ser formados pelos conjuntos de combinações dos níveis anteriores. Tem-se, dessa forma, não somente as unidades compostas, mas, principalmente, a origem da complexificação das estruturas hierarquias, bem como o seu grau de complexidade.

3.2 Sentenças como estrutura hierárquica complexa

A forma como se estabelece a comunicação entre os projetistas de produtos de peças, componentes ou sistemas mecânicos e o modelo de *PARSER* se dá através da linguagem escrita, via sentenças.

Segundo as pesquisas de Houaiss (2001, 2547), sentença é:

“sententia, ae (do latim): sentimento, parecer, opinião, idéia, maneira de ver, impressão do espírito; modo de pensar ou de sentir, vontade, desejo; opinião; sabedoria, tino, prudência; projeção, desenho; enunciado, frase, sentença, máxima; voto ou parecer...”

Tais sentenças estão estruturadas com seus constituintes - os verbos, substantivos, adjetivos, advérbios, artigos, pronomes, conjunções, preposições e interjeições. A forma como essas estruturas se apresentam, determina o grau de complexidade entre os mesmos. Em outras palavras, o grau de complexidade da sentença é definido conforme a quantidade de detalhes geométricos (*features*) modelado na peça e vice-versa. Portanto, quanto mais detalhes geométricos a peça exibir, maior será o grau de complexidade da estrutura hierárquica da peça. Nesse trabalho, a complexificação da estrutura arbórea de cada peça é diretamente proporcional à sua estrutura funcional dada sua função global, suas funções parciais ou elementares.

3.2.1 Definição de estrutura sentencial (*feature*)

Apresenta-se a seguir, algumas definições de estrutura.

Em Lalande (1999:347) *“... num sentido especial e novo, emprega-se, pelo contrário, para designar, por oposição a uma simples combinação de elementos, um todo formado de fenômenos solidários, tais que cada um depende dos outros e só pode ser o que é na e pela sua relação com eles”*.

Houaiss (2001:1267) denomina estrutura como uma *“rede de associações que se constroem a partir de correlações e oposições entre os elementos lingüísticos”*.

Para Piaget (2001:121), uma estrutura é *“... um sistema¹ apresentando leis ou propriedades de totalidade enquanto sistema. Estas leis de totalidade, em consequência, são diferentes das leis ou das propriedades dos próprios elementos do sistema”*.

Nesse trabalho, considera-se uma peça como um sistema de formas geométricas. Nesse sistema, uma rede de associações geométricas é definida como um arranjo de formas

¹ Sistema é um conjunto de estruturas de componentes que interagem de modo regular entre si e com o meio ambiente, satisfazendo a certas restrições ambientais provenientes da escassez de recursos, para atingir determinados objetivos (Perin Filho, 1995:15).

geométricas ou *features*. Os elementos ou *features* formadores dessa rede de associações geométricas são solidários entre si e cada um desses elementos pode ser escrito ou definido por uma combinação de constituintes, podendo dessa forma, determinar uma estrutura sentencial bem formada, isto é, uma sentença gramatical.

Nível	Elementos Formadores
0	<p>Letras do alfabeto : (a,b,c,d,...,z), (A,B,C,D,...,Z)</p> <p>Algarismos : (0,1,2,3,...,9)</p> <p>Sinais separadores: (i) Absolutos : ". ", "!", "?"</p> <p>(ii) Relativos: " ", " ", " ", " ", " ", " "</p> <p>Sinais especiais : "/", "@", "\$", "%", "&", "(,)", "[,]", "<, >", etc.</p>
1	<p>Tokens²: Agrupamentos de elementos do "nível 0", as cadeias de caracteres ou strings.</p>
2	<p>Sentenças ou Estruturas: Agrupamentos de tokens. Tais agrupamentos podem ser particionados em subEstruturas. Isto é, pode-se aplicar a definição de conjunto das partes de um conjunto como em Alencar Filho (1985:45).</p>

As sentenças ou Estruturas por serem formadas por tokens podem ser classificadas em sentença ou Estrutura, subEstrutura e Estrutura vazia.

- a) Sentença ou Estrutura é toda sentença aceita pelo Modelo de *Parser*, isto é, é toda sentença gramatical cuja forma normal, nesse sistema, tem os mesmos constituintes frasais correspondentes a alguma árvore hierárquica de funções de peça;
- b) SubEstrutura é toda sentença aceita pelo Modelo de *Parser*. Entretanto, essas sentenças possuíram formas especiais não correspondentes a nenhum módulo de

qualquer árvore hierárquica de funções de peça, pois, tais sentenças se referiram apenas aos nomes de peça ou alguma de suas partes;

c) Sentenças ou Estruturas vazias acontecerão quando não acontecer nenhum dos casos anteriores. Então, as sentenças ou Estruturas vazias corresponderam as sentenças agramaticais ou sentenças cuja forma não corresponderem a nenhum modelo da árvore hierárquica funcional de peça, nome de peça ou qualquer das partes de peça do domínio tratado.

Ainda no PLN da linguagem natural dos projetistas em sistemas mecânicos, as palavras classificam-se em não-terminais, terminais e vazias.

Assim como no dizer de Aho, Sethi & Ullman (1995: 13) “a cadeia contendo zero caractere ou zero token, escrita ‘ ϵ ’, é chamada de cadeia vazia”. Também no PLN todo token contendo apenas caractere espaço em branco é um token vazio.

Token terminal é todo aquele cujo último caractere do seu agrupamento é um separador absoluto ou um separador relativo. Sendo o último caso, o *Parser* precisa reconhecer os casos em que é necessário segmentar o texto a partir do separador relativo em evidência. Enquanto, os tokens não-terminais não são nem terminais e nem vazios.

3.2.2 Relações entre os elementos de uma estrutura hierárquica

Nesse trabalho, muitas foram as técnicas utilizadas nas diversas operações do processamento da linguagem natural do especialista em projeto de componentes, peças ou sistemas mecânicos e em todos os módulos constitutivos da arquitetura proposta. Assim sendo, uma síntese das relações de pertinência, inclusão e complementaridade entre *elementos X Estrutura*, *elementos X subEstrutura*, *subEstrutura X Estrutura* e *Estrutura X Estrutura* utilizadas nos três níveis categóricos – caractere, palavra e sentença –, estão elencados a seguir.

(1) Relação “membro_da”.

Nível 0

[Relação elemento X Estrutura] Se X é elemento do nível 0 e α uma Estrutura.

Então, X é elemento da Estrutura α se, e somente se, X é membro da Estrutura α .

Nível 1

[Relação subEstrutura X Estrutura] Se Y é elemento do nível 1 e β uma Estrutura.

Então, Y está contido na Estrutura β se, e somente se, Y é membro da Estrutura β .

Nível 2

[Relação Estrutura X Estrutura] Se Z é elemento do nível 2 e γ uma estrutura.

Então, Z é parte da Estrutura γ se, e somente se, Z é membro da Estrutura γ . (A

Figura 15 é um exemplo dessa relação).

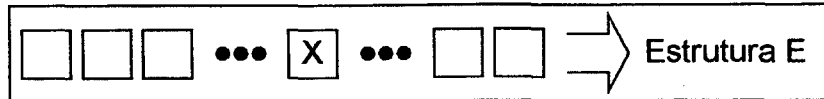


Figura 15: Elemento membro de uma Estrutura E

(2) Relação “primeiro_elemento”.

Dados um token X e uma Estrutura E, uma lista³ numérica ou alfabética qualquer.

Então, X é o primeiro elemento da Estrutura E se, e somente se, X é a cabeça da Estrutura E. (Ver Figura 16).

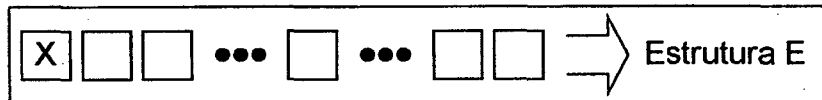


Figura 16: Primeiro elemento de uma Estrutura E

(3) Relação “sucessor_de”.

Dados dois tokens X e Y de uma Estrutura E⁴.

Então, Y é sucessor de X na Estrutura E se, e somente se, Y está imediatamente à direita de X (Cf. Figura 17).

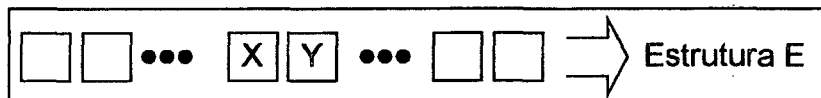


Figura 17: Elemento Y como sucessor de X na Estrutura E

(4) Relação “antecessor_de”.

Dados dois tokens Z e W, membros de uma estrutura E⁴.

Então, Z é antecessor de W na Estrutura E se, e somente se, Z está imediatamente à esquerda de W Estrutura dada (Observe a Figura 18).

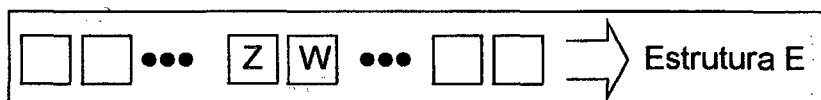


Figura 18: Z como Elemento antecessor de W na Estrutura E

³ Observe-se que uma lista, [Head|Tail], pode ser do nível 0, 1 ou 2.

⁴ uma lista numérica ou alfabética qualquer

(5) Relação “a_esquerda_de”.

Seja X um token e S_E uma subEstrutura da Estrutura E^4 .

Então, a subEstrutura S_E está à esquerda de X se, e somente se, X for um elemento sucessor de S_E na Estrutura E (Veja Figura 19).

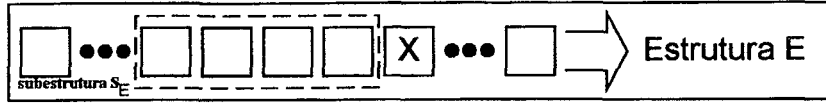


Figura 19: Elemento X à direita da subEstrutura S_E na Estrutura E

(6) Relação “a_direita_de”.

Sejam E^4 e $(S_E)^4$ uma Estrutura e uma subEstrutura, respectivamente, e X um membro da Estrutura E .

Então, a subEstrutura S_E está à direita de X se, e somente se, X for um elemento antecessor de S_E na Estrutura E (Veja Figura 20).

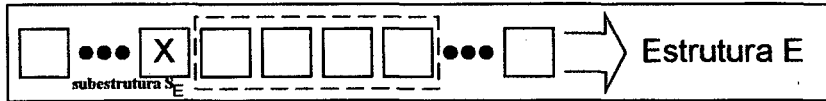


Figura 20: Elemento X à esquerda da subEstrutura S_E na Estrutura E

(7) Relação “ultimo_elemento”.

Seja a Estrutura E^4 e X um de seus membros.

Supondo que Tam é o tamanho da Estrutura E . Então, X é o último elemento da Estrutura E se, e somente se, a posição de X na Estrutura E é Tam . (Ver Figura 21).

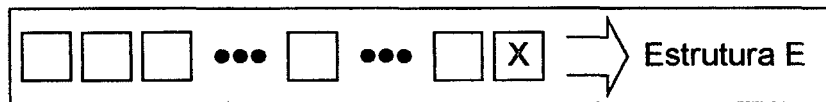


Figura 21: Último elemento da Estrutura E

(8) Relação “elementos_ordenados”

Seja a Estrutura E^4 .

Se X , Y e Z são elementos da Estrutura E . Então, os elementos da Estrutura E estarão em:

i) Ordem Crescente se, e somente se, os sucessores dos elementos dessa Estrutura, partindo-se do primeiro (da Cabeça) ao penúltimo elemento (elemento

antecessor da Calda)⁵ da Estrutura E, forem sempre maiores ou iguais que os seus respectivos sucessores (veja Figura 22);

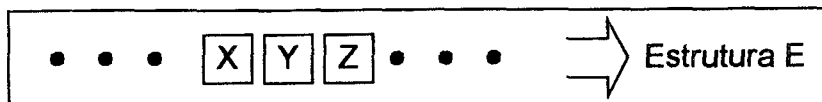


Figura 22: Elemento com ordem crescente na Estrutura
Se $X < Y, Y < Z$

ii) Ordem Decrescente se, e somente se, os sucessores dos elementos dessa Estrutura, partindo-se do primeiro (da Cabeça) ao penúltimo elemento (elemento antecessor da Calda)⁵ da Estrutura E, forem sempre menores ou iguais que os seus respectivos sucessores (veja Fig. 23);

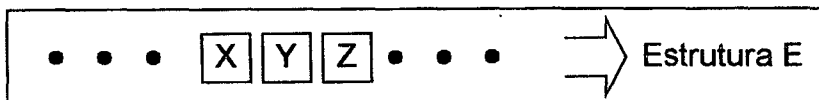


Figura 23: Elemento com ordem decrescente na Estrutura
Se $X > Y, Y > Z$

(9) Relação “concatenar_elementos”

Sejam E' e E'' Estruturas⁴ e X e A , respectivamente, membros dessas Estruturas.

Imergir E' e E'' na operação de concatenação é agrupar os elementos dessas Estruturas em uma única Estrutura E , conservando⁶ a ordem em que os elementos de ambas se encontram (Confira Figura 24).

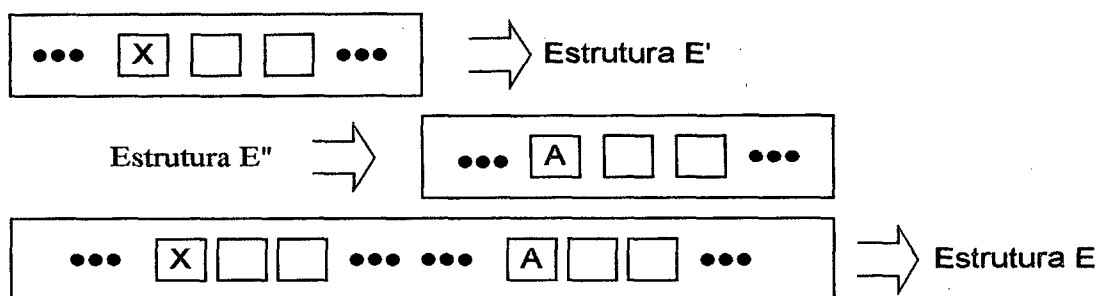


Figura 24: Concatenação dos elementos de Estruturas diferentes em uma única Estrutura E

(10) Relação “substituir_elemento”

Seja uma Estrutura E^4 e $S_{E'}$ sua SubEstrutura⁴.

Supondo $S_E = \{X, Y, W, Z\}$ e $S_{E'} = \{A, F, K, T\}$.

Então, para permutar S_E por $S_{E'}$ basta:

⁵ Não esquecer que na lista $[H|T]$ o último elemento é o elemento vazio, $[]$.

⁶ Desde de que as Estruturas E' e E'' possuam a mesma categoria estrutural (nível 0, 1 ou 2).

i) Particionar a Estrutura E de tal forma que os elementos de S_E sejam isolados;

ii) Trocar os elementos de S_E pelos elementos de $S_{E'}$;

iii) Concatenar os novos elementos da Estrutura $S_{E'}$ na Estrutura E.

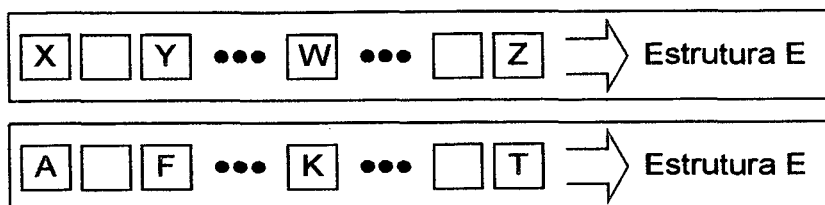


Figura 25: Substituição de um ou mais elemento na Estrutura E

(11) Relação “retirar_elemento”

Seja uma Estrutura E^4 e uma ocorrência de membro(s) repetido(s) em E (por exemplo, a repetição do elemento X em E).

Então, para retirar os elementos repetidos de E basta:

1) Particionar a Estrutura E de tal forma que os elementos repetidos fiquem isolados;

2) Separar as repetições indesejáveis;

3) Eliminar as repetições indesejáveis;

4) Concatenar os elementos restantes na Estrutura E.

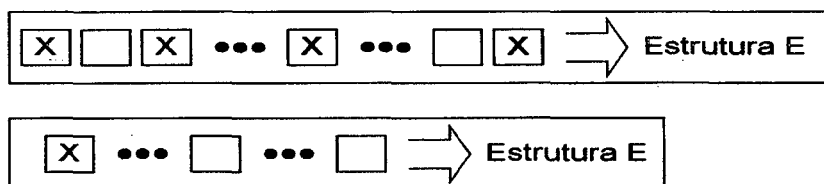


Figura 26: Retirada de elementos repetidos em uma Estrutura

(12) Relação “elementos_complementares”

Seja a Estrutura E^4 e $(S_E)^4$ sua subEstrutura, onde os elementos de S_E são definidos por uma dada propriedade.

Suponha que como resultado de uma imersão de E em dada aplicação, os elementos de S_E se tornaram exceções, isto é, elementos indispensáveis para a aplicação. Entretanto, os elementos que só pertencem a E tornaram-se indesejáveis. Nesses termos, para resgatar os elementos de S_E basta eliminar os elementos complementares a S_E em relação a Estrutura E.

Então, retirando a complementaridade da subEstrutura S_E , em relação a E (que são os elementos complementares, parte azul da operação entre os conjuntos) tem-se a parte amarela como resultada da operação da imersão, conforme exposto na Figura 27.

$$C_E^{S_E} = E - S_E, E \supset S_E = \text{Parte Amarela.}$$

Analogamente, para a operação de complementaridade entre Estruturas, conforme exposto na Figura 28, tem-se:

$$C_E^{E'} = E - E' = \text{Parte Amarela.}$$

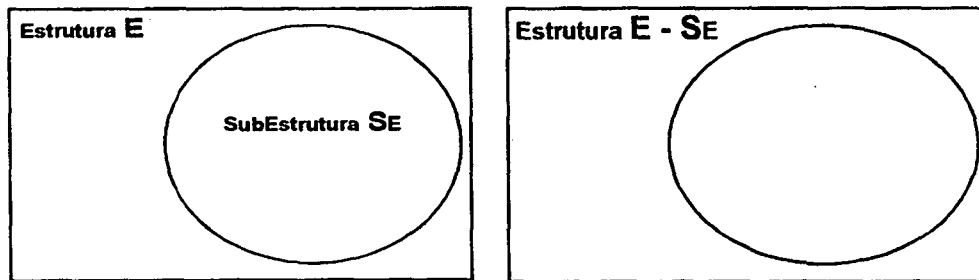


Figura 27: Operação de Complementaridade entre Estrutura e subEstrutura

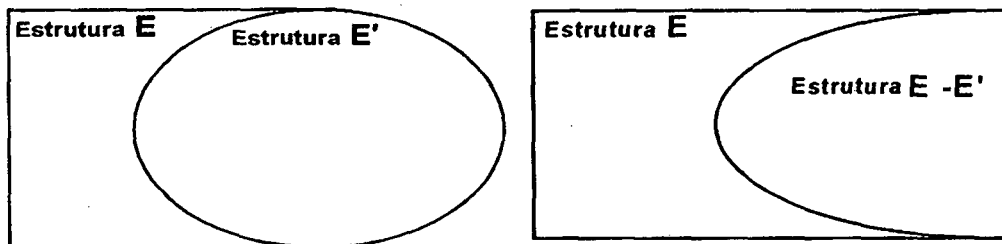


Figura 28: Operação de Complementaridade entre Estruturas

(13) Relação “agrupar_elementos”

É uma forma de conversão ascendente do nível 0 ao nível 1 ou do nível 1 para o nível 2.

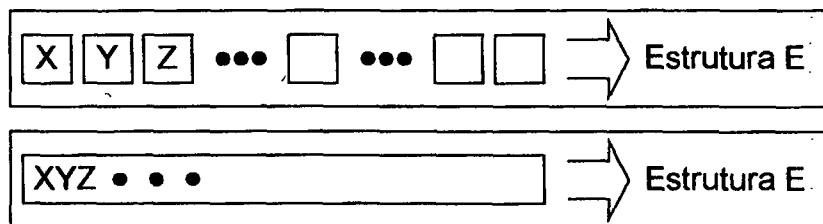


Figura 29: Agrupamento de elemento em uma única Estrutura

(14) Relação “desagrupar_elementos”

É uma forma de conversão descendente do nível 1 ao nível 0 ou do nível 2 para o nível 1.

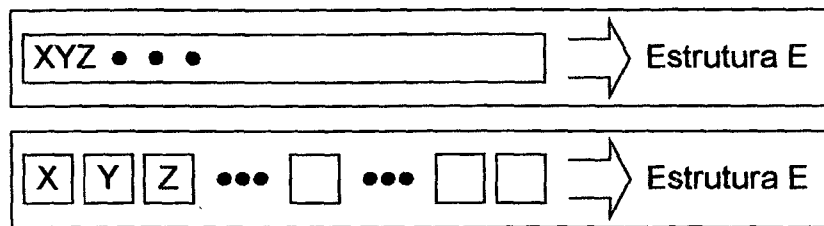


Figura 30: Desagrupamento dos elementos de uma única Estrutura

3.3 Arquitetura do modelo de *PARSER* proposto

Na Figura 31, abaixo, tem-se a proposta de arquitetura do *Parser*.

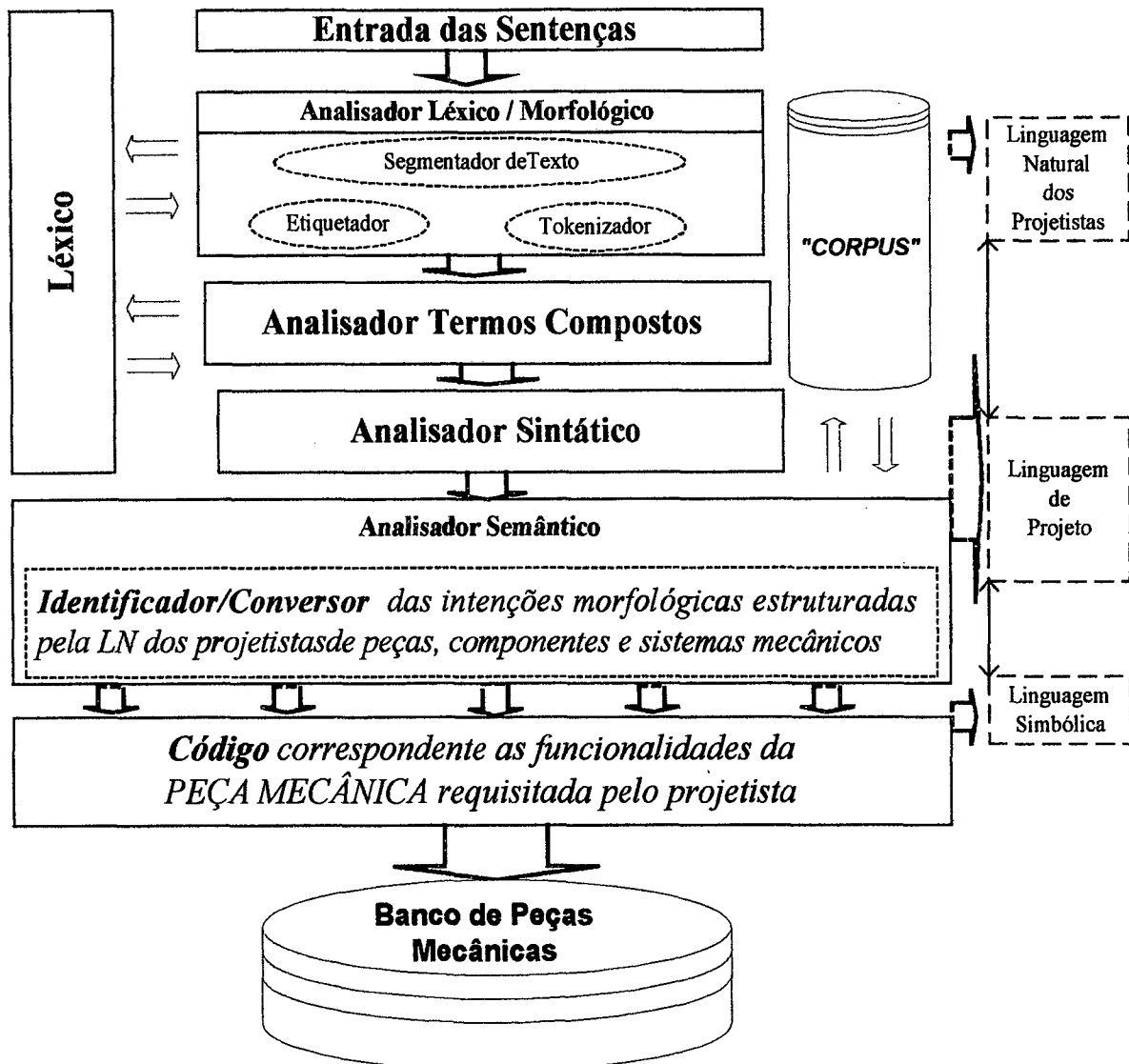


Figura 31: Arquitetura do Modelo de *PARSER* proposto

3.3.1 Módulo Segmentador de Texto

O conceito de frase, estabelecido por autores como Savadovsky, 1987: 07, Savadovsky, 1988: 03, Cegalla, 1989: 269, Faraco & Moura, 1994: 380, Mesquita, 1994: 307 e de um modo geral pela gramática normativa, na análise de texto por interfaces em LN tem sido até então a gênese para uma adequada interpretação de frases e textos no estabelecimento da comunicação entre homem-homem e homem-máquina.

Portanto, depreende-se que frase é a unidade mínima de comunicação lingüística ou frase é todo enunciado capaz de transmitir, a quem nos ouve ou lê, tudo o que pensamos, queremos ou sentimos. Pode revestir as mais variadas formas, desde a simples palavra até o período mais complexo, elaborado segundo os padrões sintáticos do idioma. Por isso que, o entendimento do sentido de frase se constitui na tarefa preliminar e imprescindível para diversas aplicações dentro da Engenharia do Conhecimento. Trata-se, pois da importância dada para a configuração do módulo segmentador de texto na determinação das sentenças (frases) pela máquina.

Logo, a tarefa do módulo segmentador de texto é dividir o texto em frases como mostra a Figura 32.

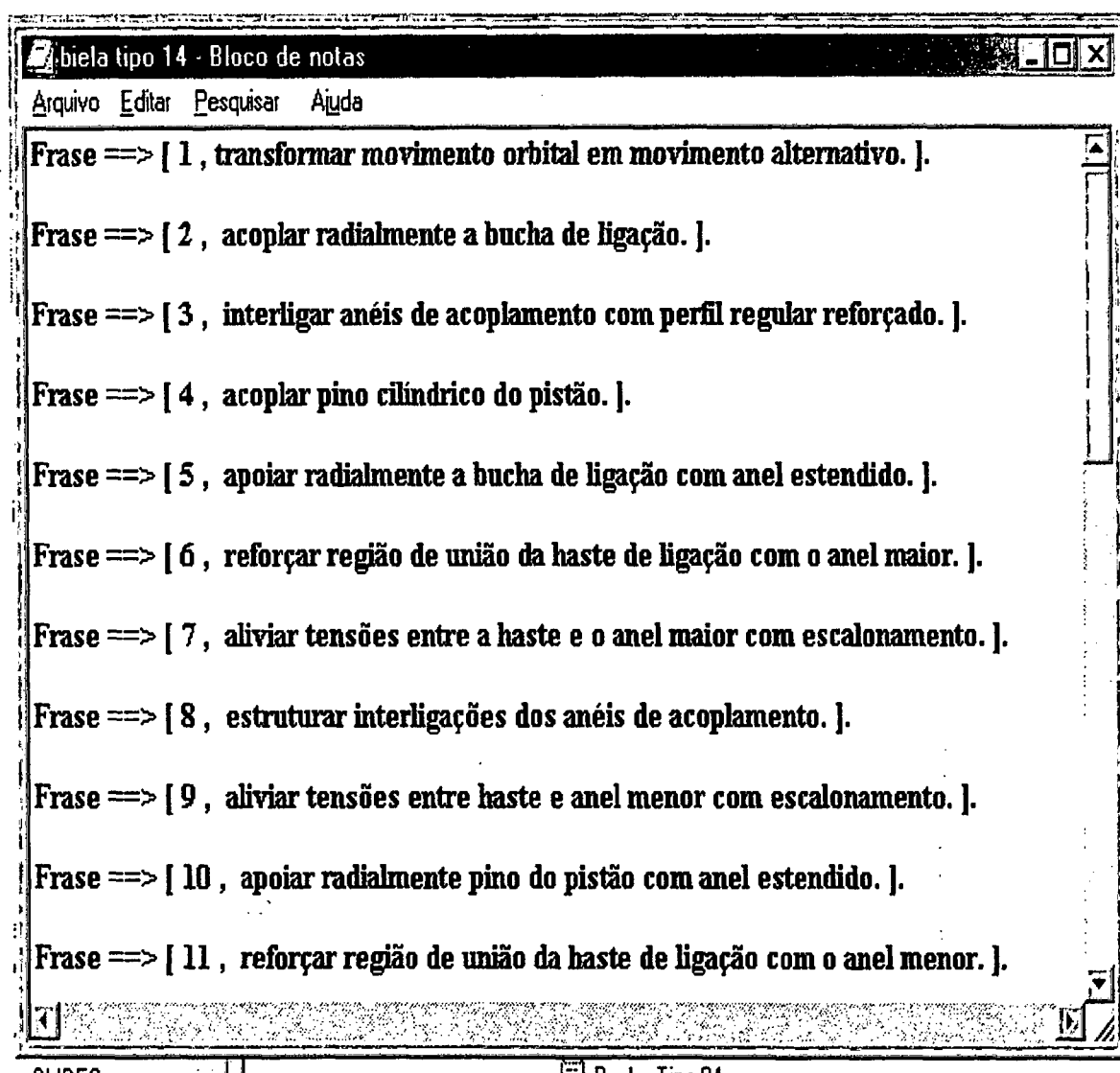


Figura 32 – Resultado da segmentação do texto biela tipo 14

Para cumprir tal tarefa, o módulo segmentador de texto deverá identificar com competência o token terminal de cada sentença pertencente ao texto. Então, identificado o ponto final num token, o segmentador de texto tem que separar imediatamente a sentença. Para isso, o segmentador de texto é obrigado a diferenciar um ponto final de um ponto pertencente a e-mail, URL, abreviatura, acrônimo, número decimais, das classes numéricas, dos ordinais, dos pronomes de tratamento, bem como de separadores absolutos de sentenças como o próprio ponto final, o ponto de interrogação e o ponto de exclamação. Além disso, o módulo segmentador de texto deverá diferenciar os casos em que os separadores relativos são tokens terminais.

Entende-se, nesse sentido, como separadores relativos o ponto e vírgula, dois pontos, reticências, parênteses, travessão único, vários sinais de pontuação juntos e sinais de pontuação contíguos. Pode-se obter mais informações sobre esse tópico no endereço eletrônico <http://cgi.portugueses.mct.pt/treebank/criteriosseparação.html> (Afonso, Susana e Marche, Ana Raquel: 2001).

No escopo desse trabalho, apenas alguns critérios de segmentação de sentença foram implementados nesse protótipo. Pretende-se, num outro momento implementar os casos de segmentação de texto ainda não contemplados.

3.3.2 Módulo Tokenizador de Sentenças

Esse módulo tem como objetivo particionar a sentença em tokens, isto é, em palavras. Assim sendo, uma de suas finalidades é filtrar a sentença desnudando-a e transformando-a, por exemplo, numa lista de strings, onde cada string ficaria isenta não só dos separadores relativos e absolutos como também de sinais como aspas simples, aspas duplas, parênteses, colchetes, chaves, barra simples, barra invertida, barra vertical, asterisco, travessão entre outros. Observe-se que, esse módulo não tem como finalidade eliminar outros sinais (ou ruídos) de quaisquer tokens, que não corroborem para o bom desempenho do programa.

Na Figura 33 tem-se um protótipo de interface, onde se implementou o segmentador, tokenizador, verificador de termos compostos e uma forma de pesquisar o layout de um componente, peça ou sistema mecânico.

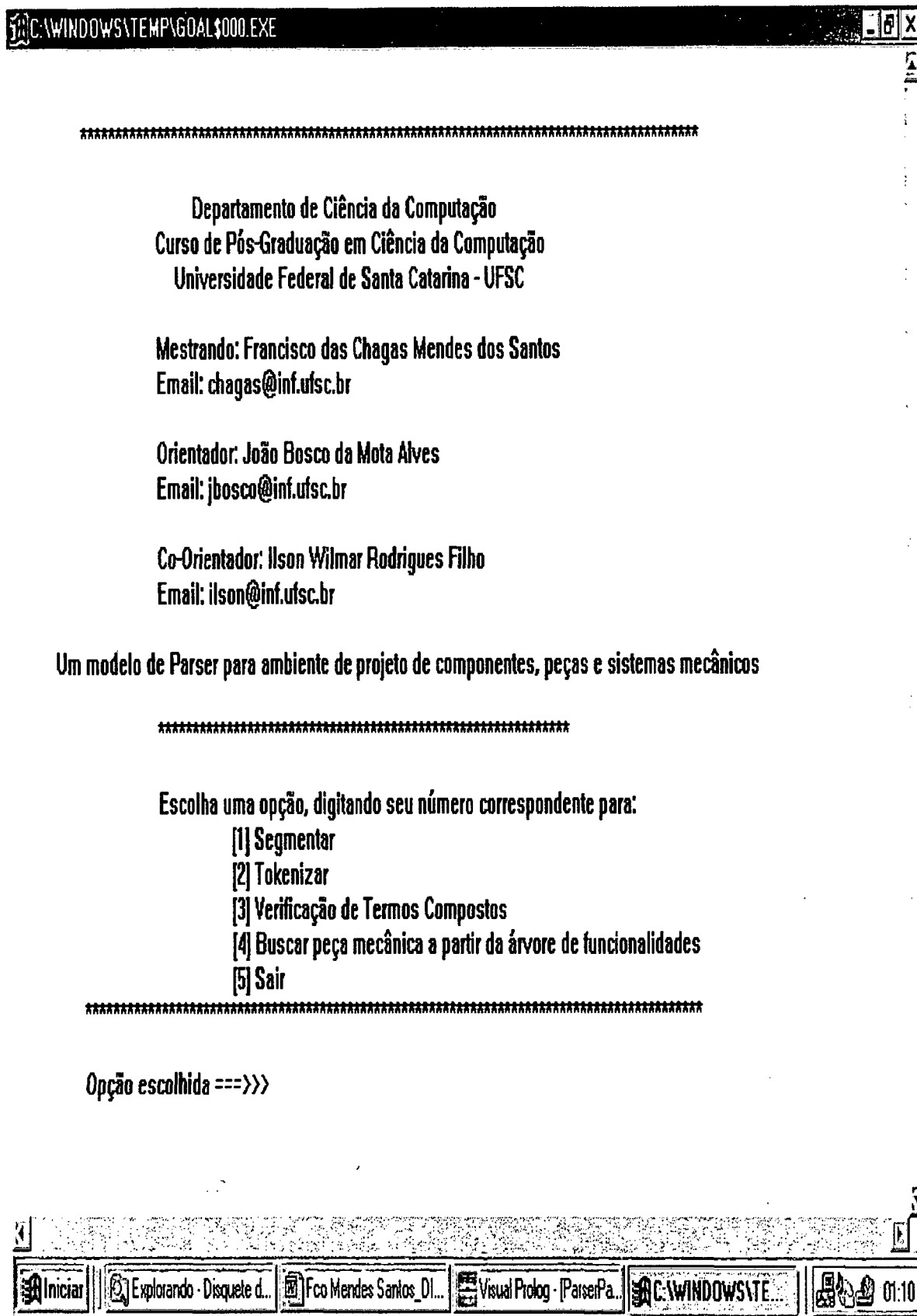


Figura 33- Protótipo com os módulos opcionais como Segmentador, Tokenizador, Verificador de Termos Compostos - Versão 1.0

3.3.3 Módulo Léxico

Segundo Perini

“O léxico é uma longa lista onde se armazena a informação idiossincrática (ou seja, não redutível a regras gerais) da língua. O léxico é composto de grande número de itens (itens léxicos), e cada um deles encerra informação sobre as características fonológicas, morfológicas, sintáticas e semânticas de uma palavra (melhor dizendo, de um lexema), de um morfema, ou ainda de uma expressão idiomática”. (2000: p.344)(grifo nosso)

Dessa maneira, as palavras (cadeias de caracteres – as strings) são as expressões básicas da linguagem. Elas são símbolos que servem para denominar objetos, propriedades dos objetos, relações entre objetos e propriedades das relações. Usualmente o termo palavra, está associado a uma forma individual (como ligação) ou a um conjunto de formas relacionadas de determinada maneira (como o par de palavras ligação/ligações).

Assim sendo, o termo palavra será utilizado para denotar cada uma das formas individuais presentes na linguagem dos profissionais de projeto de peças, componentes ou sistemas mecânicos. A entidade que engloba uma ou mais palavras dentro de uma mesma família é comumente chamada de *lexema* ou *radical*. O protótipo aqui implementado, na verdade é uma excelente base de estudo para futuras implementações de uma interface com técnicas mais avançadas ainda, como a estruturação de lexemas para palavras sinônimas.

Todavia, o conhecimento extraído de apenas um especialista de projeto de peças, componentes e sistemas mecânicos impôs nesse trabalho uma certa limitação na estruturação do léxico. Portanto, o léxico desse protótipo foi o modelado a partir de entrevistas de apenas um especialista e de outros estudos decorrentes de pesquisas. Além do mais, procurou-se realizar ainda mais estudos e pesquisas sobre um determinado módulo realizável (MR) do produto industrial compressor alternativo (objeto de estudo desse trabalho) a fim de se efetuar simulações e projetar peças pertencentes à mesma família das peças do MR desse trabalho. Para tais simulações usou-se o sistema CAD *SolidWorks 2000*.

Exemplos da estruturação de algumas palavras pertencentes ao léxico:

Palavra	Categoria	Gênero/Forma Nominal	Número
Transformar	Verbo	Infinitivo	Singular
Pistões	Substantivo	Masculino	Plural
Movimento Orbital	Substantivo	Masculino	Singular

3.3.4 Módulo Analisador Léxico/Morfológico

A função do analisador léxico é reconhecer os elementos da sentença ou texto, atribuindo-lhes categorias sintáticas ou eventualmente, cumprindo a função de análise morfológica. Isso poderia permitir que o dicionário contivesse apenas radicais de palavras, em vez de todas as formas necessárias (e adequadas para o projeto piloto do modelo do *Parser* em questão), como as correspondentes a masculino e feminino, singular e plural bem como todas as flexões verbais.

Entretanto, o módulo analisador léxico/morfológico nesse projeto se limitará a trabalhar com os traços sintáticos previamente etiquetados a partir das categorias e classes gramaticais e de acordo com o banco de dados extraído do especialista de projeto de peças, componentes e sistemas mecânicos. Os traços sintáticos ou *features* dos constituintes são os mencionados na seção 2.3.3.1.

3.3.5 Módulo Composição de Termos Compostos

Na análise léxica desse protótipo, é possível identificar palavras compostas como substantivo + adjetivo, substantivo + preposição + adjetivo. Por exemplo, pino cilíndrico, eixo excêntrico, ar refrigerado, anel maior, anel menor, anel estendido, bucha de ligação, haste de ligação, entre outros.

Essas construções também foram etiquetadas e armazenadas diretamente no dicionário da mesma forma que as demais palavras desse projeto.

O módulo de tratamento de erros ainda não foi implementado, pois julgamos ser necessário uma pesquisa mais apurada sobre as estatísticas dos possíveis erros nesse tipo de ambiente, além de uma adequada modelagem para os prováveis lexemas ou outras metodologias.

3.3.6 Módulo Analisador Sintático

O objetivo do processamento sintático, chamado de análise sintática (*parsing*, em inglês) é transformar seqüências lineares de palavras em estruturas que mostrem como elas estão relacionadas entre si, utilizando o formalismo das gramáticas livre de contexto. Isto

é, o formalismo da DCG Prolog, mas baseada nas estruturas da gramática léxico-funcional, as estrutura-c e estrutura-f.

A estrutura dos constituintes (estrutura-c) determina o tipo¹ da sentença. Ela possibilita o descobrimento do princípio de funcionamento da sentença a partir da sua análise e com base nas regras léxico-funcionais. Nessa perspectiva, a análise consiste em explicitar a organização categorial representantes dos elementos constitutivos da sentença individualmente.

Nesse trabalho, constatou-se somente a presença das sentenças do tipo declarativa, nas entrevistas com o projetista de componentes, peças ou sistemas mecânicos.

Já a estrutura funcional de cada sentença (estruturas-f), é constituída a partir das suas descrições funcionais enquanto sentenças, a partir das inter-relações entre seus constituintes. Elas (as descrições funcionais) constituem um passo intermediário entre a estrutura-c e a estrutura-f da sentença possibilitando a análise da mesma a partir das propriedades requeridas pela gramática à qual a sentença pertence.

Então, dada à configuração frasal da sentença, a análise da estrutura-f pode ser operacionalizada tendo-se como base às regras léxico-funcionais. Isto é, pelas regras de produção que associam as características, traços sintáticos, de cada palavra do léxico com as funções dos constituintes na própria sentença da linguagem natural do profissional de projeto de sistemas mecânicos.

Os formalismos gramaticais, das DCG e LFG, explorados neste trabalho são os que normalmente se usa como linguagem artificial para a representação dos recursos de uma língua. Esses formalismos foram usados na modelagem computacional dos processos de análise² e síntese³ das sentenças da linguagem coloquial do especialista em projeto de sistemas mecânicos.

Sabe-se que a DCG é baseada no conceito de ferramenta lingüística. Ela é uma representação computacional para as línguas naturais. Enquanto, a LFG foi desenvolvida a

¹ As sentenças podem ser dos seguintes tipos: obrigatórias e facultativas. Sendo obrigatórias as declarativas, interrogativas, imperativas e exclamativas. As do tipo facultativas são as negativas, enfáticas e passivas.

² Na análise o sistema verifica se a sentença é bem formada, isto é, se a sentença é gramatical ou agramatical.

³ Na síntese o sistema era buscar combinações de itens lexicais para produzir sentenças gramaticais.

partir da teoria léxico-funcional, a qual procura caracterizar as restrições e os princípios universais da linguagem.

Ambos os formalismos das DCG e LFG são baseados na GCL, ou seja, possuem a restrição de que qualquer regra de produção é da forma ‘uma variável deriva uma palavra’. Adicionalmente, um formalismo é restrito se ele possui domínio e forma bem definidos, podendo ser, portanto, bem representado e analisado.

Quanto à complexidade de representação, a LFG em relação a DCG faz-se representar de maneira mais simples pois, enquanto a complexidade daquela fica por conta do léxico, nesta ela fica por conta das próprias regras gramaticais, como por exemplo através do aumento quantitativo do tamanho do argumento das próprias regras. Nesse caso, entretanto, a complexidade depende do tamanho e do nível de dependência contextual entre os constituintes da gramática representada.

Exemplos:

a) Acoplar radialmente o anel maior da biela.

Tipo: Sentença Declarativa

A análise fica por conta da gramática:

<sentença> → sintagma_verbal, sintagma_nominal

<sintagma_verbal> → **verbo, advérbio** sintagma_nominal

<sintagma_nominal> → **determiner, substantivo**, sintagma_nominal

<sintagma_nominal> → **preposição, substantivo** | ε.

A síntese fica por conta do léxico (ver seção 3.3.3).

b) Transformar movimento orbital em movimento alternativo

Tipo: Sentença Declarativa

A análise fica por conta da gramática:

<sentença> → sintagma_verbal, sintagma_nominal

<sintagma_verbal> → **verbo**

<sintagma_nominal> → **substantivo**, sintagma_nominal

<sentagma_nominal> → **preposição, substantivo** | ε.

A síntese fica por conta do léxico (ver seção 3.3.3).

3.3.7 Módulo Analisador Semântico

A meta do módulo analisador semântico é tentar processar as informações sobre a peça requerida pelo projetista, via o conjunto das suas função global e funções parciais e elementares. Procurando cumprir essa meta, a informação sobre cada *feature* é estruturada em sentença da LN pelo projetista. Essa LN⁴ poderá ser codificada em LD⁵ pelo projetista via teclado. Assim, a intenção morfológica poderá ser encapsulada e enviada ao sistema. Então, o sistema procurará correlacionar cada uma das *features*, agora em LD e segundo o projetista, com alguma semântica disponibilizada no sistema de codificação & classificação de peças KK-3.

A semântica disponibilizada nesse sistema é totalmente dependente da taxonomia imposta pelo sistema de codificação & classificação de peças KK-3. Modelado o sistema, ele se encarregará de processar e converter cada uma das funcionalidades da peça (descrições frasais imprimidas na árvore de funcionalidades de peça) em um código, que no processamento final representará o layout da peça mecânica requerida a partir de um código final.

Entretanto, a peça exigida pode não se encontrar no banco de peças mecânicas disponibilizado. Nesse caso, o módulo semântico pode ainda ser configurado para pesquisar a existência de peças com algumas similaridades daquela requerida pelo projetista. Dessa forma, a pesquisa no banco de peças pode agrupar peças com layout que interessem a esse profissional.

A Figura 34 mostra uma interface incipiente, mas que é funcional para os objetivos propostos aqui. Ela é fruto do estudo, pesquisa e mineração incisiva de um aprendiz em Engenharia do Conhecimento. Mas não nos enganemos, pois ainda há um longo caminho a ser percorrido.

⁴ Linguagem Natural

⁵ Linguagem Digital

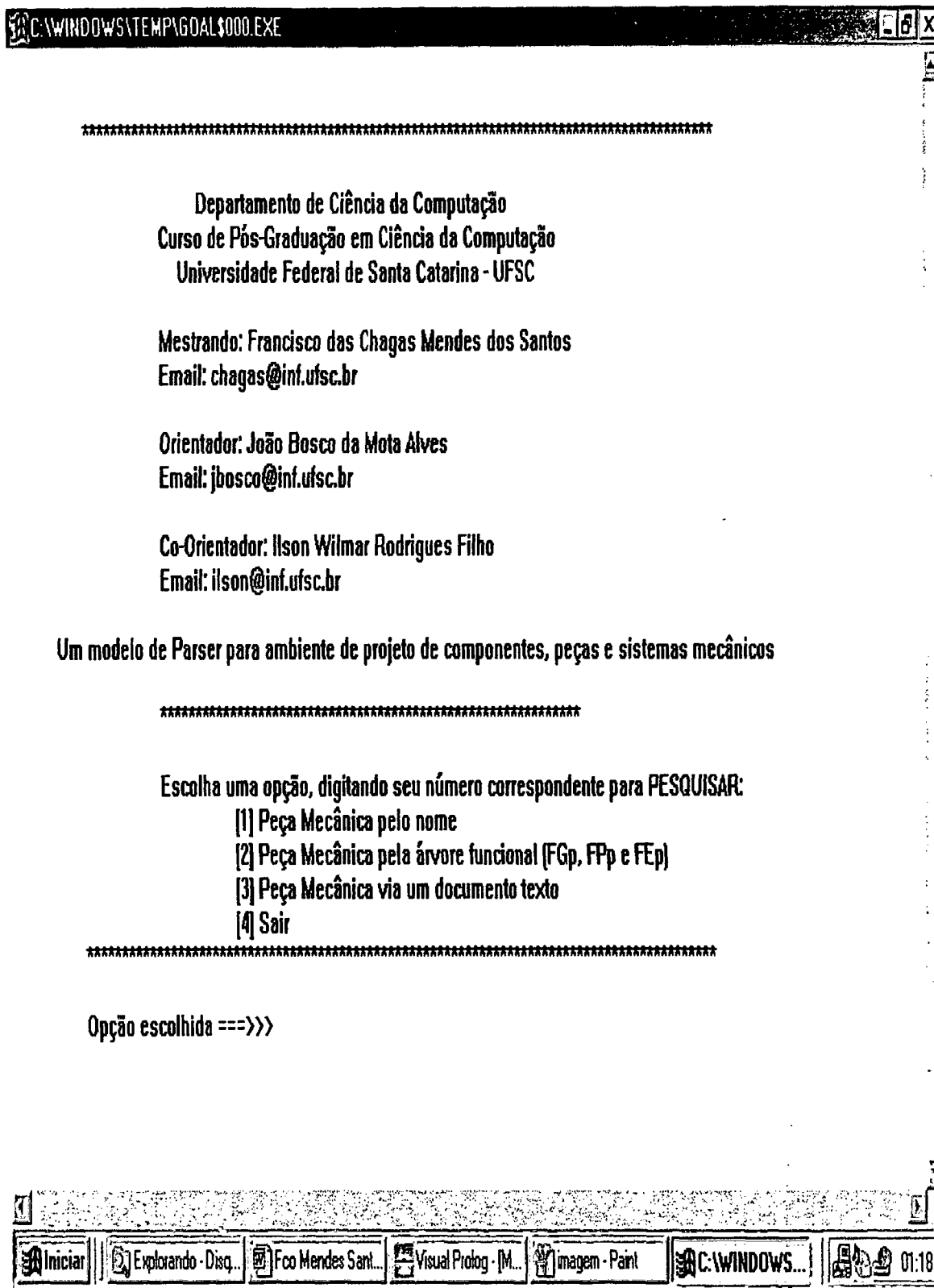


Figura 34 - - Protótipo com os módulos opcionais de pesquisa de Peças Mecânicas através do: Nome da Peça, FGp/FPps/Feps ou Árvore Funcional via texto - Versão 1.0

3.3.7.1 Sistema de Codificação & Classificação de peça KK-3

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----

Significado de cada Dígito do Código

- 1 - 2: Tipo de Peça
- 3 - 4: Tipo de Material
- 5 - 7: Dimensões da Peça
- 8 : Escalonamento
- 9 : Rosca
- 10 : Ranhuras
- 11 : Formas Irregulares
- 12 : Superfície Externa
- 13 : Superfícies Cíclicas
- 14 - 16 : Superfície Interna
- 17: Sobre a Superfície na Extremidade da Peça
- 18 : Superfícies Internas
- 19 : Furos Especiais
- 20 : Usinagem
- 21 : Tolerâncias e Precisão

O sistema codificador & classificador KK-3 possui vinte e um dígitos. Cada dígito tem uma semântica própria derivada da relação *feature*/funcionalidades de peça. Porém, esse sistema foi concebido para codificar & classificar universos de peças, isto é, ele é altamente generalista.

Evidenciou-se nesse trabalho, a necessidade de sistemas de codificação & classificação de peças mais especialista e em domínio específico.

Até o momento não se tem nenhuma proposta oficial de padronização ou qualquer tipo de sistemas de codificação & classificação de peças específicos, seja para o domínio do produto compressor alternativo, seja para outro produto qualquer. Essa é uma prática que há muito tempo perdura no céu da indústria. Comenta-se que tal prática é devido ao problema do sigilo industrial.

O fato é que a carência de dicionários especializados para domínios específicos de produtos industriais se encontra em cheque, haja vista o interesse de aplicações de sistemas processadores de linguagem natural em tais ambientes industriais.

Na Figura 35 tem-se uma amostra do processamento das informações em linguagem natural técnica de uma estrutura hierárquica funcional de dez sentenças. Isto é, aqui o sistema de processamento de linguagem natural processou a árvore de funcionalidades da biela tipo 14. Ao final do processamento ela determinou o código final representativo da peça requerida.

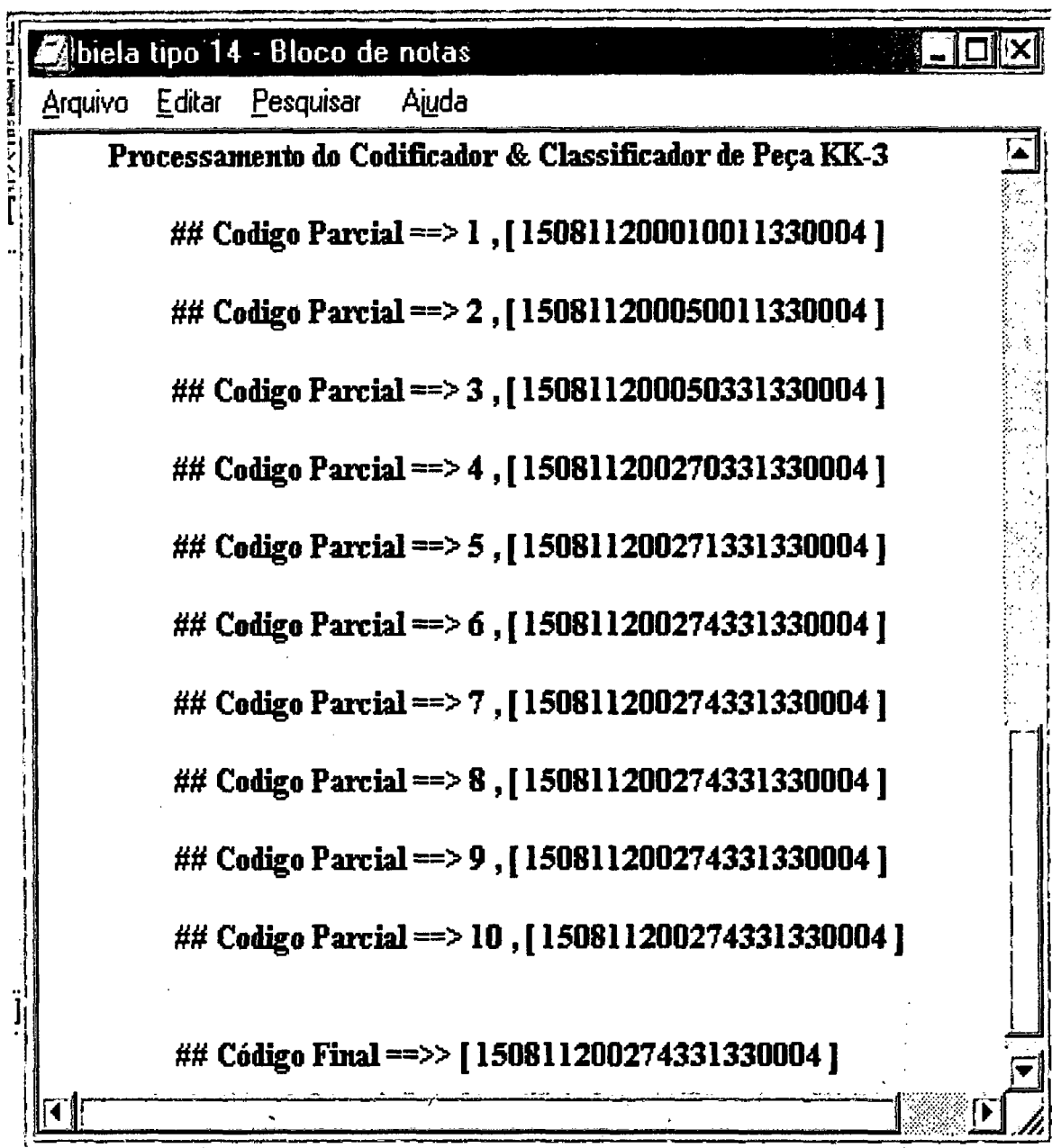


Figura 35 – Processamento da árvore de funcionalidades da peça biela tipo 14 via código KK-3

Na Figura 36 são mostradas informações básicas sobre o sistema de pesquisa de peças em um banco de dados.

Na Figura 37, tem-se a interface do banco de dados representando a biblioteca de banco de peças mecânicas de um sistema CAD. A função básica dessa interface é determinar a peça requerida pelo profissional de projeto de componentes, peças ou sistemas mecânicos a partir do correlacionamento entre código final e peça mecânica.

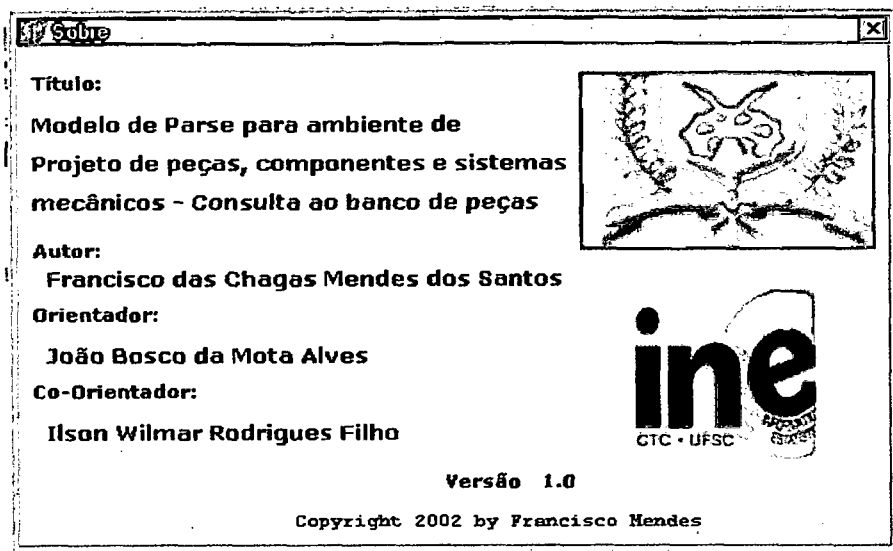


Figura 36 – Informações sobre a interface do banco de peças mecânicas

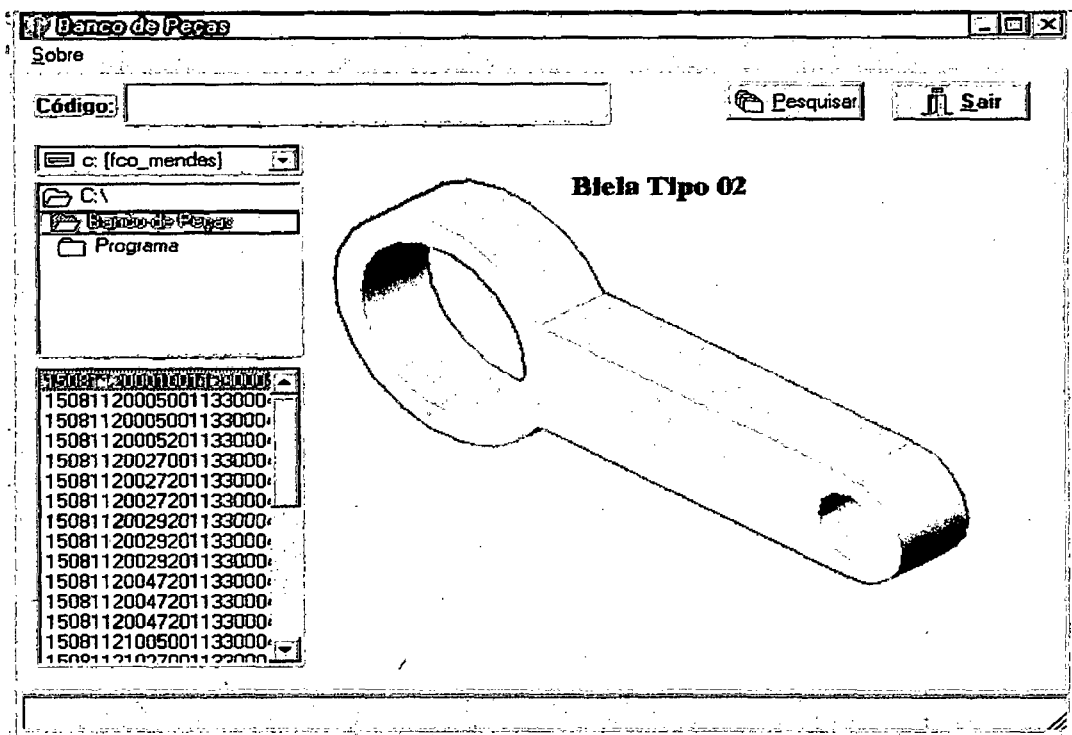


Figura 37 – Interface do banco de peças mecânicas

4 CONCLUSÕES E TRABALHOS FUTUROS

4.1 Conclusões

Esta dissertação tratou da implementação do protótipo de uma proposta de Modelo de *Parser* para auxiliar o projetista de projetos de peças, componentes ou sistemas mecânicos na sua concepção de projeto, capaz de processar a linguagem natural técnica desses profissionais nesses ambientes, a partir da função global e das funções parciais e elementares de peça.

Apesar da impossibilidade de se realizar uma pesquisa *in locus*, na origem do fenômeno - no ambiente do processo de projeto do produto - este trabalho fruto de estudos, pesquisa e mineração de dados apresenta pistas que apontam para a possibilidade de uma implementação focada no domínio do processo de projeto de qualquer produto industrial.

Esse trabalho também acena, com a possibilidade da implementação de um sistema orientada à funcionalidade para auxiliar os projetistas de peças, componentes e sistemas mecânicos. Dessa forma, o sistema aqui mencionado poderia agregar, adicionalmente às funcionalidades de um sistema CAD, um módulo inteligente orientado às funcionalidades de peça de um domínio específico, a partir de árvores de funcionalidades à luz da tecnologia *Design by feature*.

Nessa perspectiva, tal sistema poderia ser considerado uma extensão de um sistema CAD. Talvez a retomada dos estudos traçados por Hounsell & Case (1999) possam apontar para caminhos e soluções que possibilitem fundamentar sistemas CAD 'ainda mais inteligentes'. Nesses estudos, os referidos autores atestaram que o maior problema da tecnologia *Design by feature* é justamente a incapacidade de validação geométrica da sua camada semântica.

Entretanto, para viabilizarmos a implementação de um sistema orientado à funcionalidade, precisaríamos realizar pesquisas voltadas para um domínio específico tendo como base tal tecnologia, tomando ainda, como eixo principal, o que de mais promissor essas tecnologias oferecem que é precisamente a sua camada semântica¹.

Portanto, um sistema orientado à funcionalidade forneceria elementos para a comprovação da hipótese de que existem, na linguagem coloquial do projetista,

¹ Camada semântica é um conjunto de informações usadas para descrever uma peça.

regularidades ou traços lingüísticos, próprios do contexto do processo de projeto e com uma taxonomia própria. O referido sistema configuraria uma forma geométrica espacial, dotada de uma topologia na qual cada região superficial da peça pudesse tomar forma quadrangular, triangular, circular ou outras formas geométricas realizáveis para a determinação do *layout* da mesma, bem como de suas interfaces com outras peças ou até mesmo com outros MR do produto.

4.2 Trabalhos Futuros

1º) Estruturar um sistema de construção de peças mecânicas a partir do paradigma orientado à funcionalidade de peça. Esse paradigma teria como meta, correlacionar atributos funcionais com atributos geométricos de forma a possibilitar a geração das categorias funcionais de peça como as FGp, FPp e FEp.

Para isso, necessitaríamos a partir de um domínio específico:

- Realizar estudos e pesquisas ainda mais consistentes;
- Fundamentar o paradigma orientado à funcionalidade objetivando:
 - ☞ Identificar e classificar os constituintes frasais;
 - ☞ Definir os principais marcadores sintáticos;
 - ☞ Diferenciar os marcadores sintáticos em função da relação forma geométrica/especificidade de uso/Produto;
 - ☞ Mapear regras construtivas que originem os princípios de solução de peça;
 - ☞ Propor uma taxonomia para criação de um sistema codificador & classificador de peças no domínio selecionado.

2º) Elaborar uma interface amigável com foco em um domínio específico contendo:

- Um dicionário especializado e robusto para ambiente de projeto de componentes, peças e sistemas mecânicos a partir de uma pesquisa mais elaborada;
- Um corretor automático para o dicionário proposto;
- Implementação de lematizadores² para correção automática.

² Lematizador é uma operação que dada uma palavra, determina outras correspondentes.

3º) Módulos complementares:

- Verificar a possibilidade da aplicação de RNA na correção automática da LN a ser manipulada pelo PLN nesse ambiente, principalmente quanto à eficiência para a aplicação em foco;
- Estudar a potencialidade de uma taxonomia para criação de um sistema de codificação & classificação de peça específico para o domínio de aplicação. Ex: no domínio dos compressores alternativos;
- Investigar a necessidade de um processador semântico especializado para o ambiente de projeto de sistema mecânico ou outro domínio de pesquisa, voltado justamente para a camada semântica da tecnologia *Design by feature*;
- Estudar uma forma de integração mais adequada do PLN a um determinado Sistema CAD, a luz da tecnologia *Design by feature* e a partir do paradigma das funcionalidades, isto é, da função global e das funções parciais e elementares da peça, num domínio específico e possível;
- Estudar a necessidade da implementação de um etiquetador morfossintático³ para a LN quando da aplicação no PLN no ambiente de projeto de peças, componentes e sistemas mecânicos escolhido.

³ Morfossintático que tem ao mesmo tempo relação com a morfologia e a sintaxe.

REFERÊNCIAS BIBLIOGRÁFICAS

- AFONSO, S.; MARCHE, A. R. **Critério de Separação de Texto**. <http://cgi.portugueses.mct.pt/treebank/criteriosseparação.html> (acessado em Janeiro/2002) .
- ALENCAR FILHO, E. **Teoria elementar dos conjuntos**. 20 ed., São Paulo: Nobel, 1990.
- ARARIBÓIA, G. **Inteligência artificial: um curso prático**. Rio de Janeiro: LTC – Livros Técnicos e Científicos, 1988.
- AHO, A. V.; SETHI, R.; ULLMAN, D. **Compiladores: princípios, técnicas e ferramentas**. Trad. Daniel de Ariosto Pinto, Rio de Janeiro: LTC – Livros Técnicos e Científicos, 1995.
- BACK N.; FORCELLINI, F. **Apostila sobre projeto conceitual**. UFSC/CTC/, 1998.
- FONSECA, A. J. H. **Sistematização do processo de obtenção das especificações de projeto de produtos industriais e sua implementação computacional**. 2000, 180f. Tese (Doutorado em Engenharia de Produção), Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.
- FORTKAMP, M^a B. M.; TOMITCH, L. M^a B. **Aspectos da lingüística aplicada: estudos em homenagem ao professor Hilário Inácio Bohn**, Florianópolis: Insular, 2000.
- HOUAISS, A.; VILLAR, M. S. **Dicionário Houaiss da Língua Portuguesa**. 1 ed., elaborado no Instituto Antonio Houaiss de Lexicografia e Banco de Dados da Língua Portuguesa, Rio de Janeiro: Objetiva, 2001.
- HOUNSELL, M. S.; CASE, K. **An intent-driven paradigm for feature-based design**. (to appear at the) 15th International Conference on CAD/CAM Robotics & Factories of the Future (CARS&FOF'99): Águas de Lindóia, SP, Brazil, August, 1999.
- IRANI, S.A.; KOO, H. Y.; RAMOS, S. **Feature-based operation Sequence Generation in CAPP**. *International journal of production research*. v.33, n.1, p. 17-39, 1995.
- LALANDE, A. **Vocabulário técnico e crítico da filosofia**. Trad. Fátima S. Correia et all. São Paulo: Martins Fontes, 1999.
- LE, V. T. **Techniques of prolog programming: with implementation of logical negation and quantified goals**. Singapore: Jonh Wiley & Sons, inc, 1993.

LINHARES, J. C. **Modelamento de dados para o desenvolvimento e representação de peças: estudo de casos**. 2000, 183f. Dissertação (Mestrado em Engenharia de Produção), Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.

LÉVY, P. **As tecnologias da inteligência: o futuro do pensamento na era da informática**. Trad. Carlos Irineu da Costa. Rio de Janeiro: Ed. 34, 1997.

MAYER, R. J. et al. Ecotof: a *feature* representation technique for concurrent engineering applications. **Journal of design and manufacturing**. v.33, nº1, p.49-65, 1994.

MIOTO, C.; SILVA, M^a E. F.; LOPES, R. E. V. **Manual de sintaxe**. 2 ed., Florianópolis: Insular, 2000.

MONARD, M^a. C.; NICOLETTI, M^a. C. **Programas Prolog para processamento de listas**. Versão 2.0, <http://labic.icmc.sc.usp.br/portuguese/courses.html> (acessado em Janeiro/2002) .

NIVETTE, J. **Princípios de gramática gerativa**. Trad. Nilton V. da Gama. São Paulo: Pioneira, 1996.

NONAKA, I.; TAKEUCHI, H. **Criação de conhecimento na empresa: como as empresas japonesas geram a dinâmica da inovação**. 2 ed., Trad. Ana B. Rodrigues & Priscilla M. Celeste, Rio de Janeiro: Campus, 1997.

OLIVEIRA, I. L. **Uma abordagem conexcionista para resolução de anáforas pronominais**, 1997, 72f. Dissertação (Mestrado em Ciência da Computação), Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.

PAHL, G.; BEITZ, W. **Engineering design: a systematic approach**. 2 ed., SPRINGER-Verlag, London, 1996.

PALAZZO, L. A. M. **Introdução à programação PROLOG**. Pelotas: EDUCAT, 1997.

PERIN FILHO, C. **Introdução à simulação de sistemas**. Campinas/São Paulo: UNICAMP, 1995.

PERINI, M. A. **Gramática descritiva do português**. 4 ed., São Paulo: Ática, 2000.

PIAGET, J. **Seis estudos de psicologia**. Trad. Maria A. M. D'Amorim e Paulo S. L. Silva. 24^a ed., Rio de Janeiro: Forense Universitária, 2001.

PAHL, G.; BEITZ, W. **Engineering design: a systematic approach**. 2.ed. Great Britain: Springer-Verlag London Limited, 1996.

PROLOG DEVELOPMENT CENTER A/S, **VISUAL PROLOG** - versão 5.0: Language Tutorial, Copenhagen: DK, 1997.

RICH, E.; KNIGHT, K. **Inteligência artificial**. Trad. Maria C. S. R. Ratto, 2 ed. São Paulo: Makron Books, 1993.

RIBEIRO, H. C. S. **Introdução aos sistemas especialistas**. Rio de Janeiro: LTC – Livros Técnicos e Científicos, 1987.

RODRIGUES FILHO, I. W. **Curso de processamento da linguagem natural**. Notas de aula. Florianópolis: Universidade Federal de Santa Catarina, 2000.

_____. **Um modelo de busca de informação usando estruturas hierárquicas complexas**, 2000, 113f. Tese (Doutorado em Engenharia de Produção), Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.

SANTOS, F. C. M.; RODRIGUES FILHO, I. W.; ALVES, J. B. M. Um modelo de *PARSER* para aplicação em ambientes de Projeto de Sistemas Mecânicos. In: **V Oficina de Inteligência Artificial**. Escola de Informática/Grupo de pesquisa em Inteligência Artificial/UCPEL. Pelotas: 21-22 de Novembro de 2001.

SALOMONS, O. W.; VAN HOUTEN, F. J. A. M.; KALS, H. J. J. Review of research in *feature-based design*. **Journal of manufacturing systems**, v.12, n°2, p.113-132, 1993.

SCARUFFI, P. **Inteligência artificial: guia para o programador**. Trad. Renato P. Casquilho, Lisboa: Presença, 1990.

SHAH, J. J. et al. The AA.S.U. *Feature testbed*: an Overview. **ASME computers in engineering conference**, v.1, p.233-241, 1990.

SOUZA E SILVA, M. C. P.; KOCH, I. V. **Linguística aplicada ao Português: sintaxe**. 9 ed., São Paulo: Cortez Editora, 2000.

TEIXEIRA, J. F. **Mente, Cérebro & Cognição**. Petrópolis, RJ: Vozes, 2000.

APÊNDICE A

Trabalhos parciais decorrentes do processo de Mestrado:

SANTOS, F. C. M.; RODRIGUES FILHO, I. W.; ALVES, J. B. M. Um modelo de *PARSER* para aplicação em ambientes de Projeto de Sistemas Mecânicos. In: **V Oficina de Inteligência Artificial**, Escola de Informática/Grupo de pesquisa em Inteligência Artificial/UCPEL. Pelotas: 21-22 de Novembro de 2001.

_____. Um sistema especialista para ambiente de projeto de peças, componentes e sistemas mecânicos. In: **II SEPEX. II Semana de ensino, pesquisa e extensão da UFSC**. Florianópolis, junho de 2002.

_____. Proposta de uma arquitetura de *PARSER* para aplicação em ambientes de Projeto de Sistemas Mecânicos. In: **VI Oficina de Inteligência Artificial**, Escola de Informática/Grupo de pesquisa em Inteligência Artificial/UCPEL. Apresentação prevista para 17-18 de Outubro de 2002.

APÊNDICE B

ALGUMAS ÁRVORES FUNCIONAIS DE PEÇAS MECÂNICAS

Biela Tipo 01



Biela Tipo 02

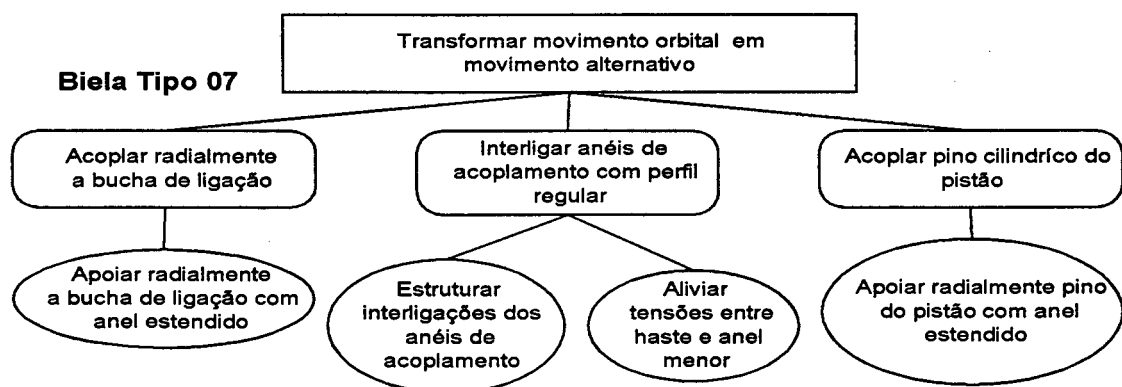


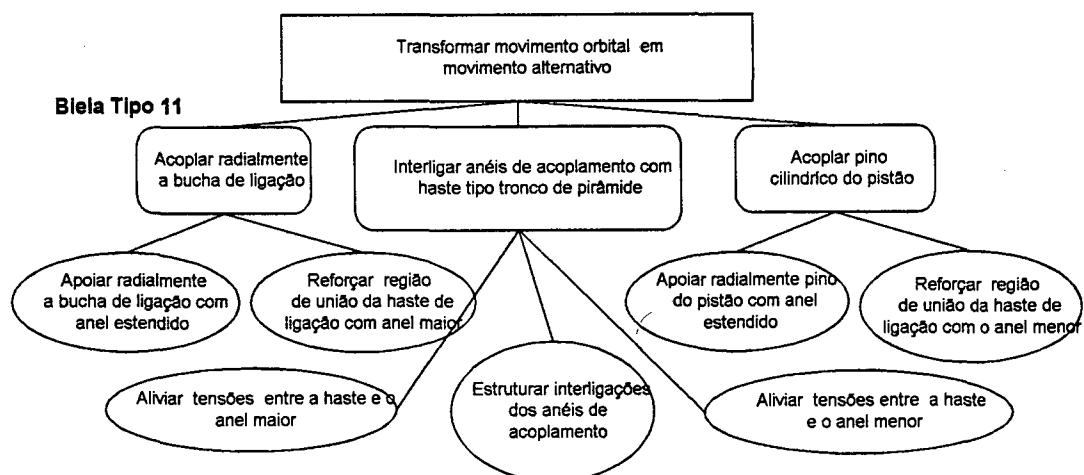
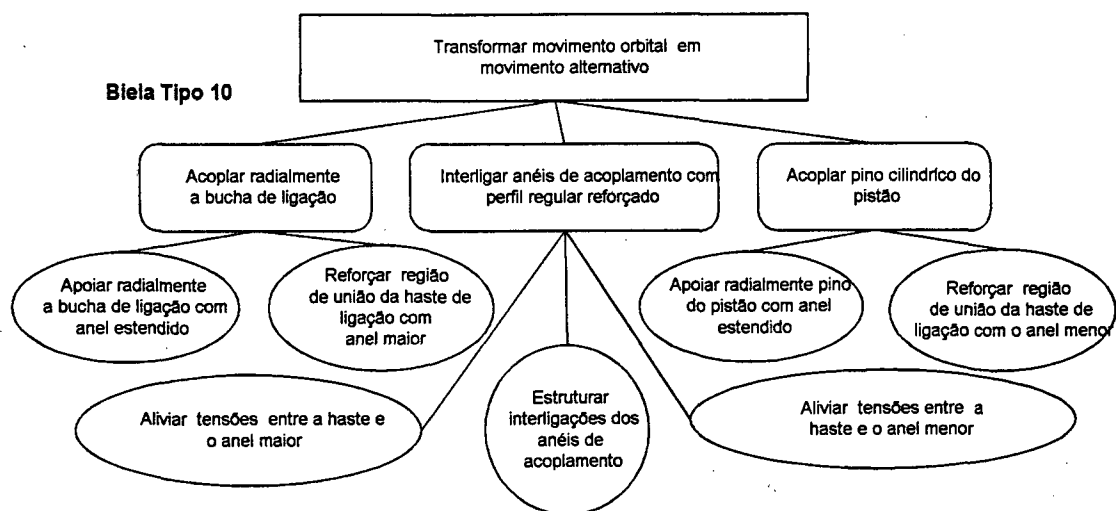
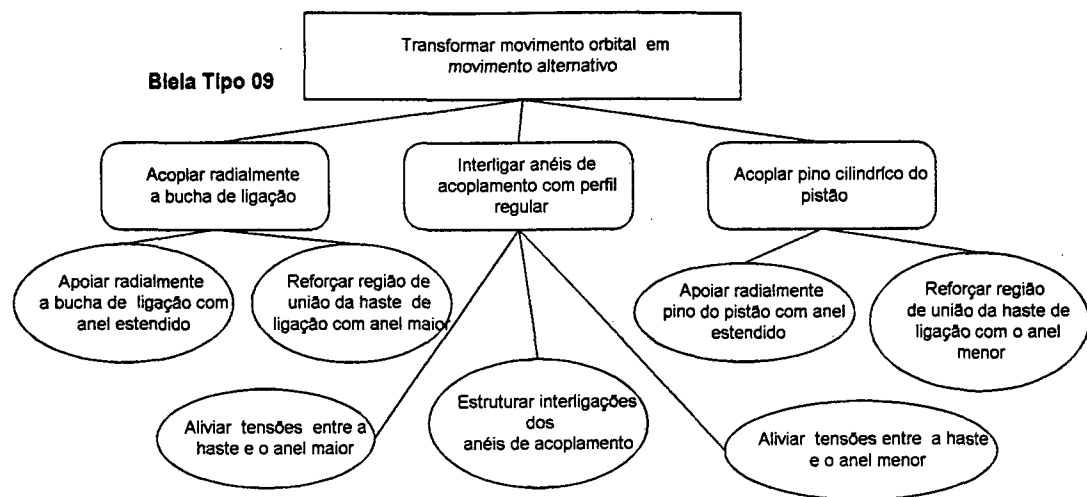
Biela Tipo 03



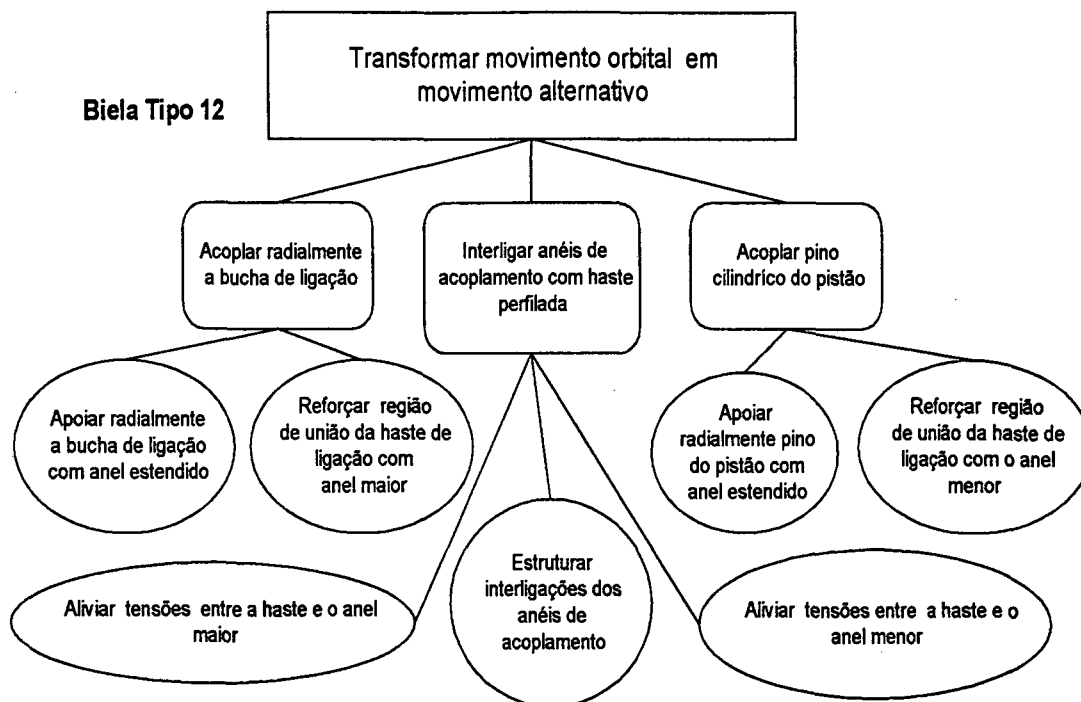
Biela Tipo 04



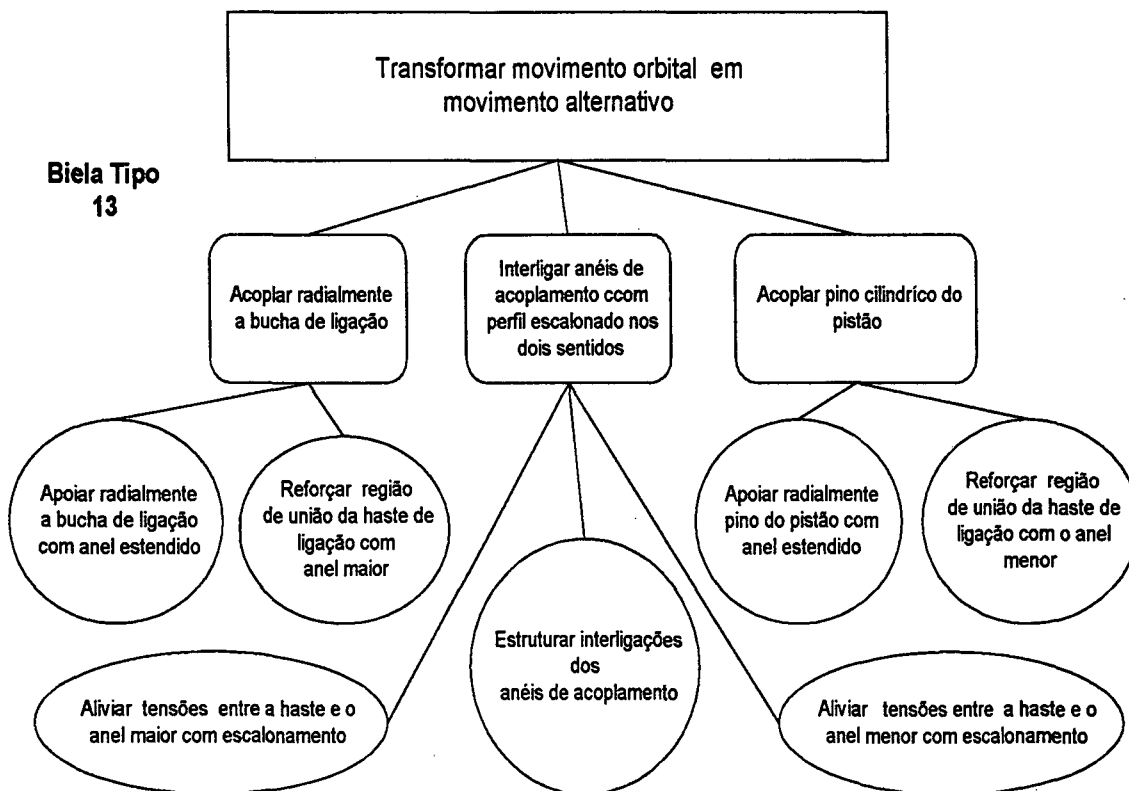


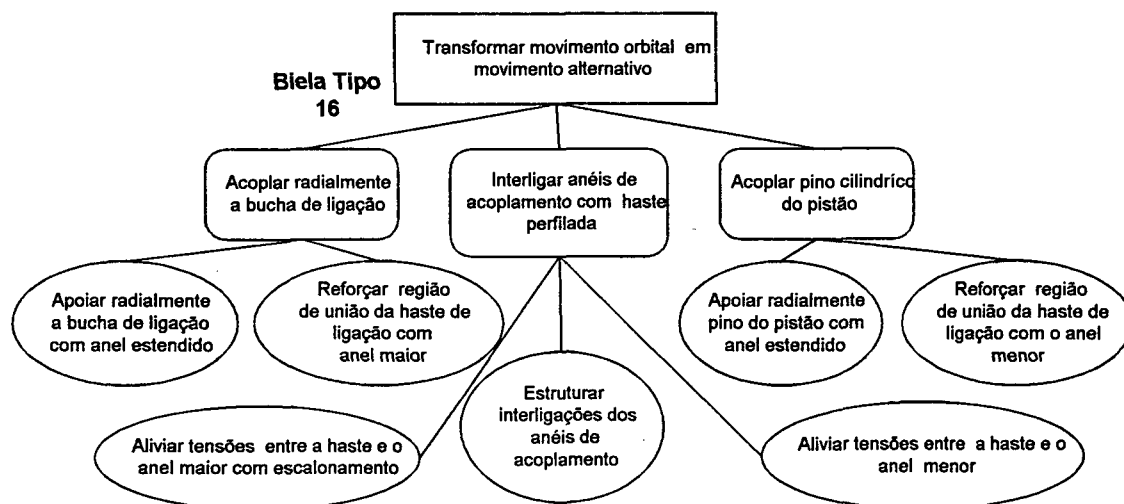
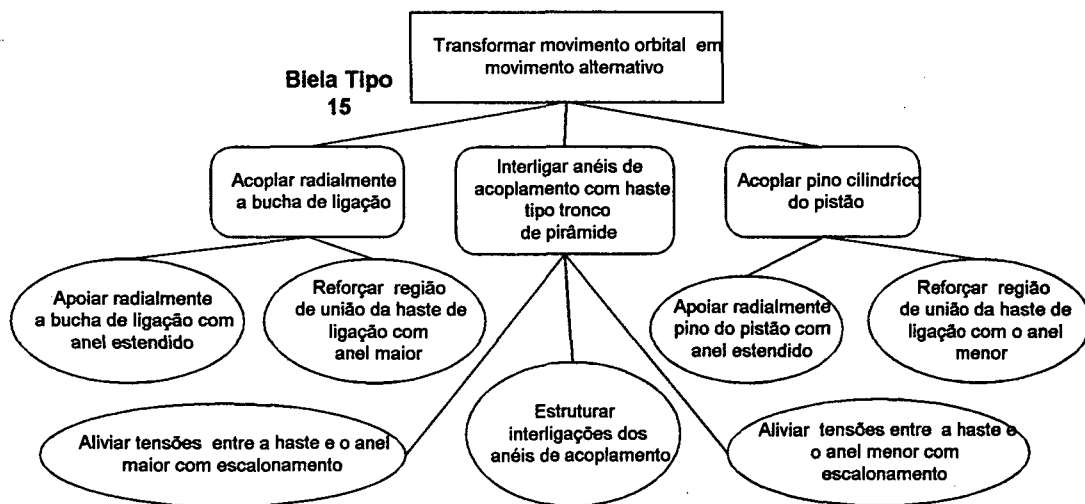
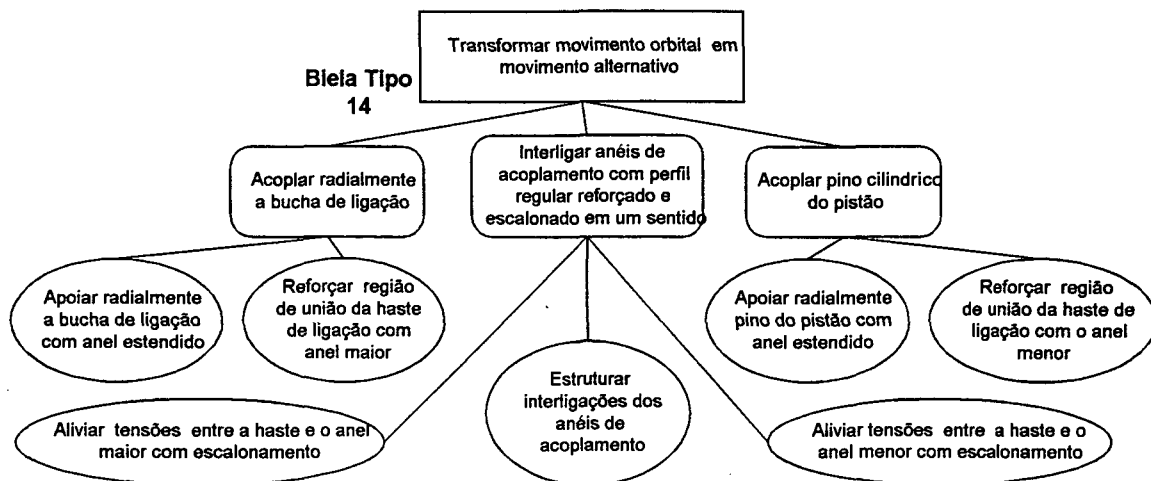


Biela Tipo 12

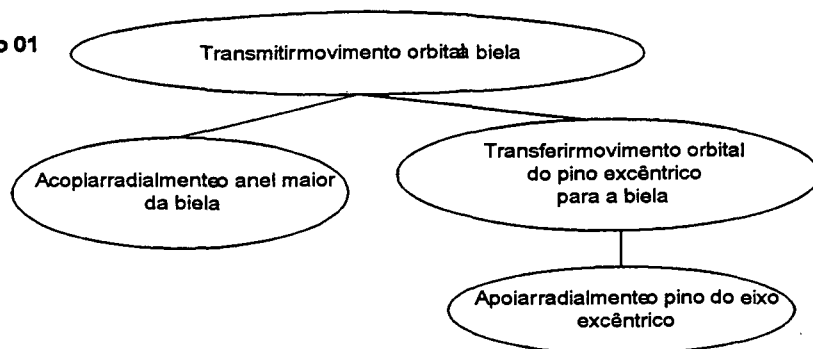


Biela Tipo 13

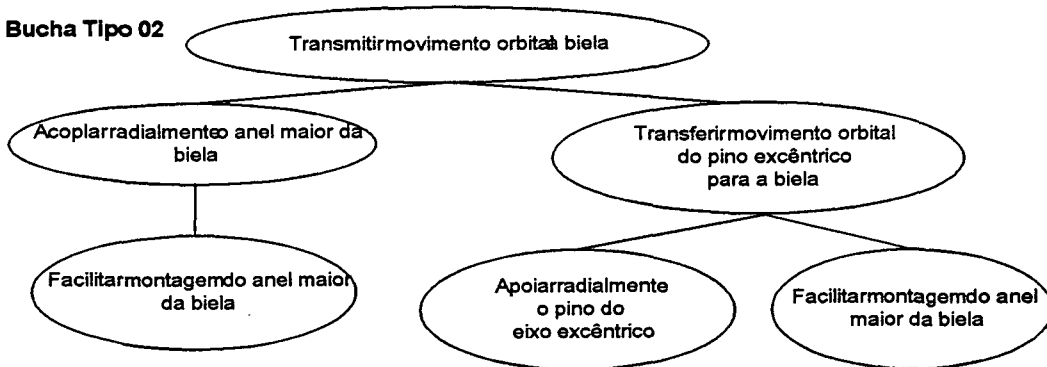




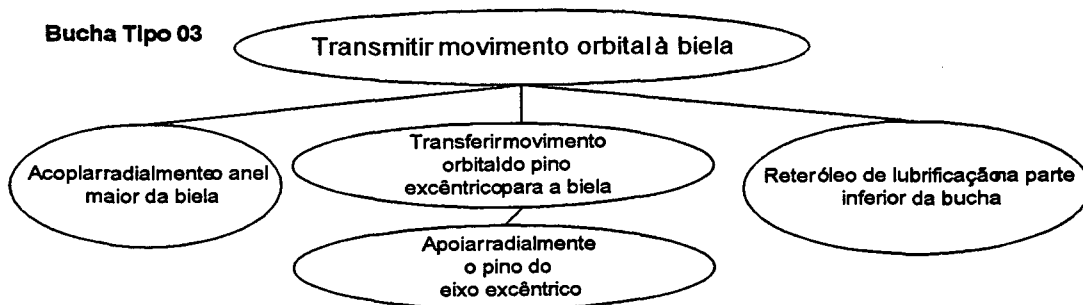
Bucha Tipo 01



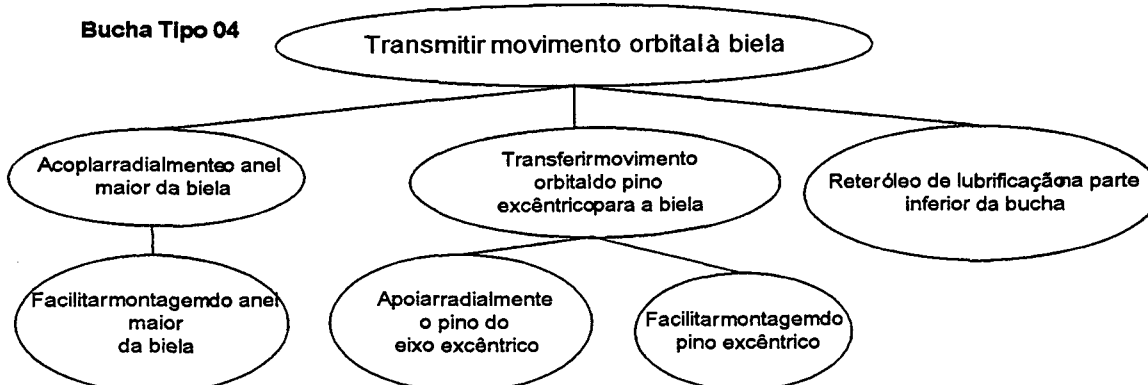
Bucha Tipo 02



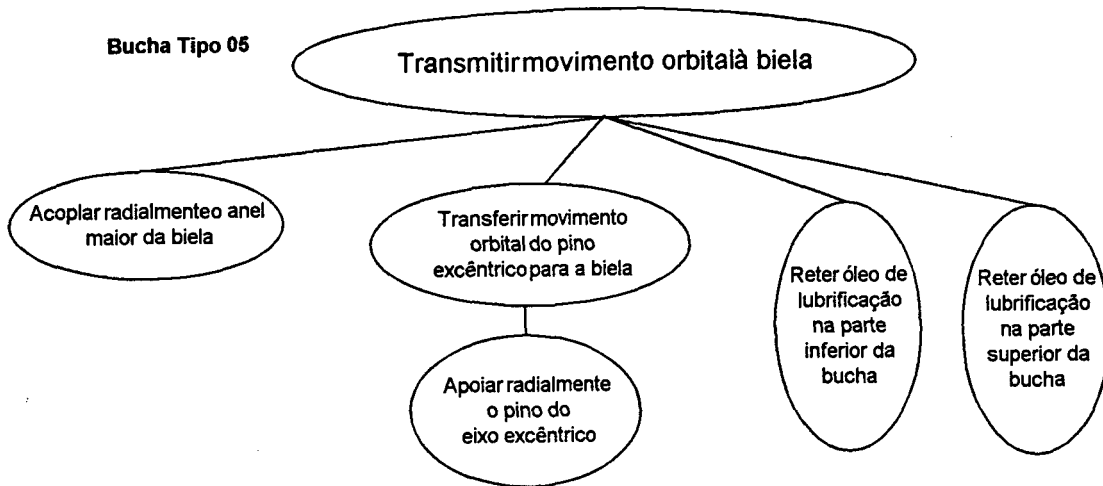
Bucha Tipo 03



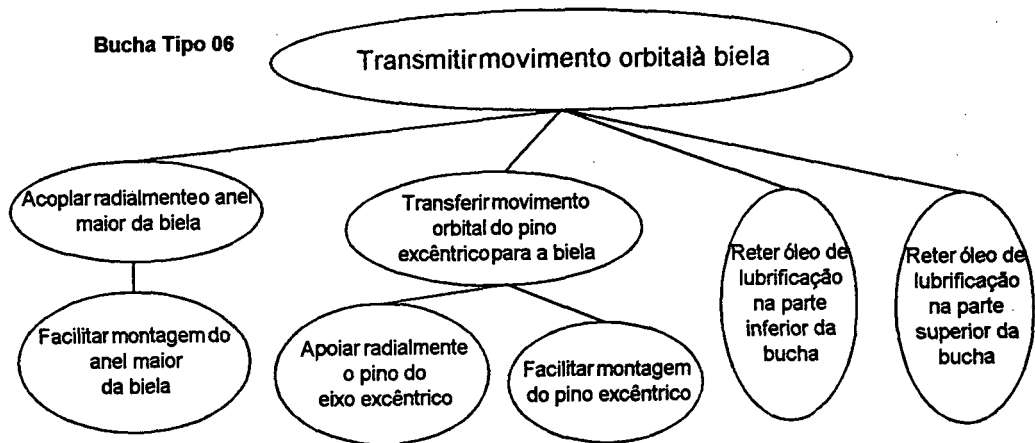
Bucha Tipo 04



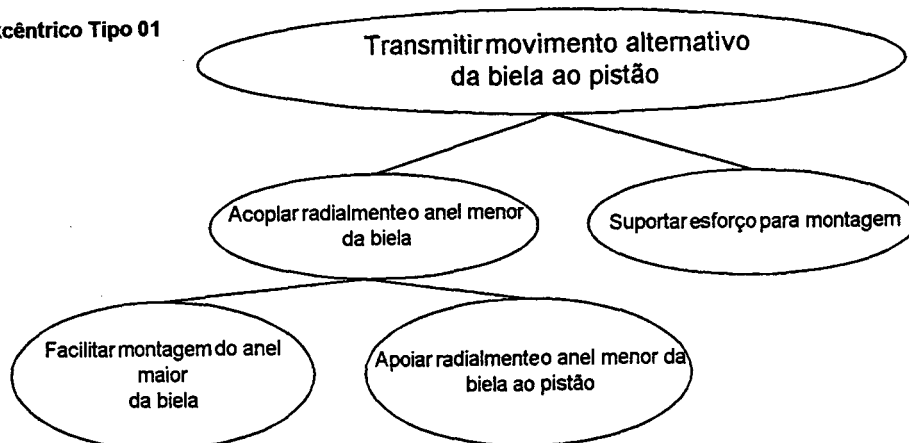
Bucha Tipo 05



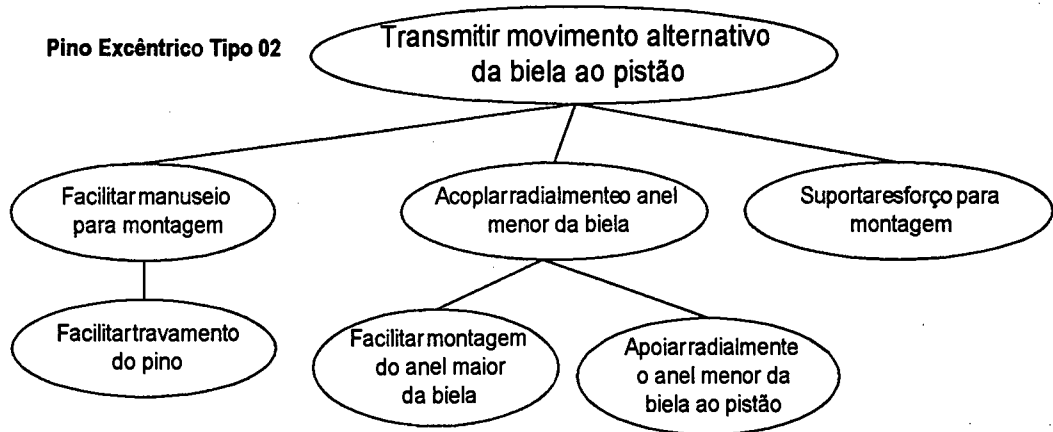
Bucha Tipo 06



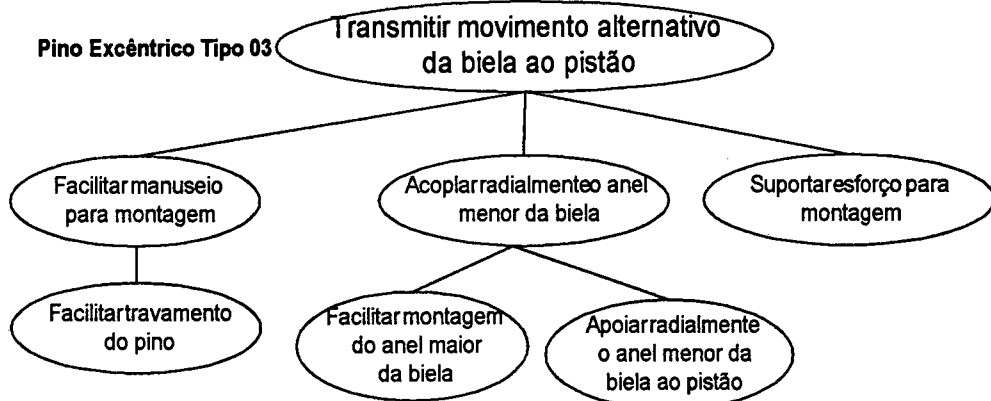
Pino Excêntrico Tipo 01



Pino Excêntrico Tipo 02



Pino Excêntrico Tipo 03



Pistão Tipo 01

